EXHIBIT A

Valtrus Innovations Ltd.'s Proposed Constructions and Intrinsic and Extrinsic Evidence

I. U.S. Patent No. 6,728,704

Patent	Claim(s)	Term	Valtrus's Proposed Construction	Examples of Intrinsic Evidence	Examples of Extrinsic Evidence
'704	1, 12	Ordering of the method steps	The steps do not all need to be performed in order.		
'704	1, 12	scoring value(s)	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'704	1, 12	search engine(s)	A computer program designed to seek out information based on a query from a user via a Web browser.	'704 Patent at 1:38-41: "Search engines are computer programs designed to seek out information based on instructions from the user. Typically, the user enters a set of instructions, often called a query, which instructs the search engine to search for specific types of information." '704 Patent at 1:21-31: "Searches for information in the networked computer environment may be cumbersome due to the sheer amount of information stored, or due to the complexity of finding information in large file structures. Indeed, with the advent of the World Wide Web (WWW) as well as other	"Search Engine." The New Oxford American Dictionary 1529 (Erin McKean ed., 2nd ed. 2005). VALTRUS-GOOGLE-NDTX-00007588-00007606. "Search Engine." Steven M. Kaplan, Wiley Electrical and Electronics Engineering Dictionary 687-88 (2004). VALTRUS-GOOGLE-NDTX-00007435-00007455. "Search Engine." Harry Newton, Newton's Telecom Dictionary 822 (24th ed. 2008). VALTRUS-GOOGLE-NDTX-00007499-00007516.

				forms of computer networking, and the corresponding explosion in the amount of information available, it is now simply impractical for users to search for information manually. The ability of search engines to analyze enormous amounts of data and isolate useful information thus becomes of paramount importance." '704 Patent at 1:52-2:33: "Almost all search engines work in this general manner. However, their architectures vary according to the context in which they operate. Search engines are currently constructed in at least three architectures: federated, peer-to-peer, and meta-search engines. Each is used to conduct different types of searches. Federated search engines are used in the client-server environment. A client or server may initiate a search for data located in various networked servers. Federated search engines are most commonly used in the WWW context, but need not be limited in this manner. Typical federated engines search the WWW by utilizing programs called bots or spiders to examine the content of information available on other computers and build an index consisting of the words or other data stored in these computers, as well as where they are located. Once users enter a query	Tao Yang and Apostolos Gerasoulis, Web Search Engines: Practice and Experience, 2 *in* Computing Handbook (3rd ed. 2014). VALTRUS-GOOGLE-NDTX-00007182-00007202. Julia Kerr, *What is a Search Engine? The Simple Question the Court of Justice of the European Union Forgot to Ask and What It Means for the Future of the Right to be Forgotten*, 17 CHI. J. INT'L L. 217, 220 (2016). VALTRUS-GOOGLE-NDTX-00007342-00007369. Sergey Brin and Lawrence Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine (1998). VALTRUS-GOOGLE-NDTX-00007266-00007285. Shaon Tewari, *How Search Engine Works*, 2 INT'L J. OF RES. IN ENGINEERING, SCIENCE & MANAGEMENT 92, 92 (2019). VALTRUS-GOOGLE-NDTX-00007293-00007301. F. Anklesaria et al., The Internet Gopher Protocol (a distributed document search and retrieval protocol) 10 (1993). VALTRUS-GOOGLE-NDTX-00007902-00007917.

				consisting of words or data desired, the search engine searches its index for any locations that contain these words/data and returns a list of such locations. The result list returned is normally a list of each such returned location and any associated information, and may include Uniform Resource Locators (URLs) for finding WWW-based data, or other expressions of data location. The results or entries in this list are often ranked according to any of a number of criteria currently available, with the goal of presenting the most relevant results to the user first. One flaw in this type of search engine is the potential for inaccurate information. Because the WWW is so large, indices are updated only sporadically, meaning searches may not uncover the most recent information. Other types of search engines avoid the need for spiders and indices, and thus present users with up-to-date information more often. One example is the peer-to-peer search engine, which can also be used for other networks besides the WWW. These search engines operate in the peer-to-peer environment, where computers are simply linked together with no centralized servers and no distinct clients. They typically work by distributing a search to various peer	Wei Tang, Search Engine Survey 1 (1999). VALTRUS-GOOGLE-NDTX-00008456-00008464. Jeffrey R. Bach et al., Virage image search engine: an open framework for image management (1996). VALTRUS-GOOGLE-NDTX-00008465-00008477. Odej Kao and Gerhard R. Joubert, A content based Internet search engine for analysis and archival of MPEG-1 compressed newsfeeds (2002). VALTRUS-GOOGLE-NDTX-00008491-00008494. Chee-Hong Chan et al., Automated Online News Classification with Personalization 7, *in* 4th International Conference on Asian Digital Libraries (2001). VALTRUS-GOOGLE-NDTX-00008445-00008455.

				computers, each of which can in turn farm out the search to other computers in the same network. In this way, individual computers search only the current contents of a few peers and not the entire WWW or other network. This eliminates the need to build a large index, and delivers to the user a real-time snapshot of the content of the network or the WWW. Finally, web meta-search engines can operate in either the client-server or peer-to-peer environment. These search engines typically act as aggregators that farm a WWW search out to other public web search engines, then process the results." '704 Patent at 3:34-52: "FIG. 2 illustrates processing operations in accordance with an embodiment of the invention. FIG. 3 illustrates an example of merging multiple result lists into a single list in accordance with an embodiment of the invention. Like reference numerals refer to corresponding parts throughout the several views of the drawings. FIG. 1 illustrates a generalized computer network 5 that may be operated in	

				accordance with an embodiment of the present invention. This computer network 5 may operate in a client-server, peer-to-peer, or other configuration, and may also be considered a representation of the WWW. The network 5 includes at least one computer 10 connected by transmission channel 20 to a group of computers 30 and 50. Transmission channel 20 may be any wire or wireless transmission channel." FIGS. 1, 2, and 3. '704 Patent at 3:52-65: "Computer 10 is a standard computer controlled by a Central Processing Unit (CPU) 12 and connected to the rest of the computers in network 5 by network connection 14. Computer 10 also includes a memory 16 that can be any form of computer-readable memory. Memory 16 contains a browser program 17 that allows users to browse the WWW. The memory 16 may also contain a search engine program 18 and an associated merging program 19 for merging different result lists, however in a client-server configuration the search engine is often resident on a different computer. The search and merging operations may be performed on any computer within the network 5." '704 Patent at 4:20-49: "The present invention operates within a network of	

				computers such as those shown in FIG. 1. More specifically, the present invention operates by engaging multiple search engines to process a query and merge the result lists for presentation to the user. In a typical client-server configuration, a user operating client computer 10 sends queries through transmission channel 20 to search engine 40, which is resident on server 30. Through the use of spiders or bots whose operations are known in the art, the search engine 40 typically will have already built up a collection of locations (which can include URLs), along with the data contained in those locations, in index 42. For example, the bots would have already searched the contents of memory 36 of computers configured like computer 30. They would have also searched the contents of WWW data pages 60 and databases 62 of computers configured like computer 50. The content from these computers would be stored in index 42. Search engine 40 then cross-checks the words or other data contained in the query against the data contained in index 42 for matches. Locations in index 42 containing data that matches the query are compiled into a result list. Search engine 40 typically transmits the same query to other search engines resident on computers configured like computer 30 and	

				connected to transmission channel 20. These other search engines then perform separate searches in the same manner as above, compile their own result lists, and return these lists to the computer 30 that originated the search. The end result of the above is a set of result lists that must be merged by merging program 44 and returned to computer 10 for display to the user." '704 Patent at 5:1-15: "Search engine 18 then performs two different tasks: it searches other computers on the network for data satisfying the query, and distributes that query to other search engines on the network 5. Here, search engine 18 searches the contents of memory 36 of peer computer 30, as well as memory 56 of peer computer 50, for data matching the query. A result list containing the locations of relevant data is then compiled. Search engine 18 also farms the same query out to the search engine 40 of peer computer 30, which conducts a search in similar fashion, examining the contents of peer computers like computers 30 and 50 and compiling the results into a list. Note that this process could continue recursively, with search engine 40 farming out the same query to other search engines in network 20, which in	

				turn could farm the search out to other search engines, and so on." '704 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000093-00000097: "Web search engines (WSE) use tools ranging from simple text-based search to more sophisticated methods that attempt to understand the intended meanings of both queries and data items." '704 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000098-00000126. '704 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000158-00000167: "Many sources on the Internet and elsewhere rank the objects in query results according to how well these objects match the original query." U.S. Patent No. 6,006,225, cited on face of '704 Patent, at 1:17-20: "With the increasing popularity of the Internet and the World Wide Web, it is common for on-line users to utilize search engines to search the Internet for desired information." U.S. Patent No. 6,102,969, cited on face of '704 Patent, at 6:43-7:19: "FIG. 2B	

				generally illustrates the user display from another example of a web-browser-based user interface embodiment, which in this case, is directed to an information domain of WWW indexes or search engines. This display is also generally divided into three sections. Section 71 displays a title for the netbot; section 72 displays the status of the current search; and section 73 displays the search results. In more detail, the display of section 71 includes a logo for this netbot, "MC" standing for "MetaCrawler," a name chosen since WWW search engines are also known as "web crawlers," and controls to access certain. system level presentation features, such the MetaCrawler home page and user feedback pages. The display of section 72 includes list 74 of the search engines being queried identified by their common names, the status of the current query in general and at each search engine, and common user controls. Generally, pie-chart icon 78 summarizes that 7 of the 8 search engines queried have already responded to the query. At search engine 75, known as "Lycos," the check mark indicates that a response containing information items has already been received. At search engine 76, known as "Inktomi," the cross mark indicates that a response without any information	

				items has already been received. On the other hand, search engine 77, known as "Galaxy," is visibly distinguished from the other search engines to indicate that it has not yet responded to the query. The common controls of section 72 include more-button 79 to request the display of newly arrived search results, and modify-search-button 80 to request a new or modified query be sent. Lastly, the display of section 73 includes the information items returned from the search engines. Each information item is displayed separately and includes title 81, descriptive text 82 if available, and line 83 with the URL of the web page for this item and the estimated relevance of this item to the query, here "1000."The items are sorted for display by descending values of the estimated relevance. The displayed items are scrolled using controls provided by the web browser. This user interface is implemented as a Java applet downloaded from a netbot server and executed by the web-browser. In this manner, the interface of FIG. 2B is capable of greater interactivity than that of FIG. 2A. For example, it can poll the netbot server for current search status and update the status displays accordingly without user action."	

				U.S. Patent No. 6,327,590, cited on face of '704 Patent, at 1:41-63: "For individual search engines, there are many different techniques for ranking results, ranging from counting the frequency of the appearance of the various search terms in the search query to calculating vector similarities between a search term vector and each returned document vector. In a networked environment such as the World Wide Web, meta-searchers access different and often heterogeneous search engines and face the additional difficulty of combining the ranking information returned by the individual engines. Meta-searcher is a Web information retrieval system aimed at searching answers to a user's query in the heterogeneous information providers distribute over the Web. When a meta-searcher receives responses (usually in the form of HTML files) from the information providers, a special component of a meta-searcher called a wrapper, process the responses to answer the original query. Since many search engines, including meta-searchers, hide the mechanism used for document ranking, the problem of merging search results is compounded. A problem common to both individual search engines and meta-search engines is that these approaches ignore, or	

				knowing nothing about, the user conducting the search, or the user's context for conducting the search." U.S. Patent No. 6,546,388, cited on face of '704 Patent, at 1:61-2:9: "Users of the Web use tools to help find, location or navigate through the Web. These tools are known as Internet search engines or simply search engines. Almost all search engines provide graphical user interfaces (GUIs) for boolean and other advanced search techniques from their private catalog or database of Web sites. The technology used to build the catalog changes from site to site. The use of search engines for keyword searches over an indexed list of documents is a popular solution to the problem of finding a small set of relevant documents in a large, diverse corpus. On the Internet, for example, most search engines provide a keyword search interface to enable their users to quickly scan the vast array of known documents on the Web for the handful of documents which are most relevant to the user's interest."	
'704	1, 12	representative value	A value that represents the scoring values of the entries of a list.	'704 Patent at 3:1-19: "The invention includes the step of transmitting a query to a set of search engines. Any result lists returned from these search engines is received, and a subset of entries from	"Representative." Collins English Dictionary 1372 (7th ed. 2005). VALTRUS-GOOGLE-NDTX-00007568-00007587.

				each result list is selected. Each entry in this subset is assigned a scoring value according to a scoring function, and each result list is then assigned a representative value according to the scoring values assigned to its entries. A merged list of entries is produced based upon the representative value assigned to each result list. The invention further includes a computer-readable memory to instruct a computer to merge multiple result lists from search engines. Executable instructions stored in the memory include instructions for selecting a subset of entries from each result list. Each entry in the subset is assigned a scoring value according to a scoring function. Each result list is assigned a representative value based on a function of scoring values assigned to its entries. The entries are then ranked based on the representative value assigned to their result list." '704 Patent at 5:65-6:5: "The next processing step is to determine, for each list, a representative score of all scoring values determined for its entries (block 78). The representative score may be the arithmetic average or a value proportional to the average for a set of scoring values. The present invention	"Representative." The New Oxford American Dictionary 1437 (Erin McKean ed., 2nd ed. 2005). VALTRUS-GOOGLE-NDTX-00007588-00007606.

				includes the step of determining this representative score according to any number of known techniques." '704 Patent at 7:10-14: "Result lists 110, 112, 114 can now be merged into merged list 140. As above, this is accomplished using the representative value assigned to each list. In this example, the representative value assigned to each list is an average scoring value." '704 Patent at 7:14-34: "In this example, the list with the highest average scoring value is selected first. In FIG. 3, this is result list 112, having an average scoring value 132 of 14.95. The first unselected entry, 1B, is selected first. Average scoring value 132 is then decremented by some amount. If that amount is chosen to be 1.0, average scoring value 132 takes on a value of 14.95−1.0=13.95. Because 13.95 is still the highest average scoring value, 2B is chosen next and average scoring value 132 is decremented by another 1.0 to take on a value of 12.95. Now, the highest average scoring value is value 134, or 13.225. Entry 1C is thus the next entry selected. Scoring value 134 is then decremented to 12.225; value 132, which is at 12.95, is now the highest value again. Entry 3B is thus chosen	

				next, and value 132 is decremented to 11.95. This means value 134 is now the highest value. Entry 2C is then chosen, and value 134 is decremented to 11.225. This means value 132 is again the highest value, so entry 4B is selected next and value 132 is decremented to 10.95. Value 130 is now the highest value, so entry 1A is chosen and value 130 is decremented to 11.25−1.0=10.25. This process repeats until all entries from all three lists are selected." 7:35-54: "According to another embodiment, each list 110, 112, 114 is assigned a probability value equal to its average scoring value's percentage of the total of all average scoring values. Entries are then selected from each list based on its probability value. Here, for instance, the total of all average scoring values 130, 132, 134 is 11.25+14.95+13.225=39.425. This means result list 110 is assigned a probability value equal to (11.25/39.425)100%=28.54%. In like manner, result list 112 is assigned a probability value of (14.95/39.425)100%=37.92%, and result list 114 is assigned a probability value of (13.225/39.425)100%=33.54%. Result lists are then selected in pseudorandom fashion, where at each selection result list 110 has a 28.54% chance of being	

				picked, list 112 has a 37.92% chance, and list 114 has a 33.54% chance. Once a list is selected, the first entry that has not already been selected is picked. Once every entry in a list is selected, the total of all average scoring values is recalculated without that list's average scoring value, and the process continues until every entry of every list has been selected." '704 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000098-00000126: "The representative of a database indicates approximately the contents of the database." '704 Patent Prosecution History, VALTRUS-GOOGLE-NDTX-00000306-00000315: "In essence, the result lists are merged with the goal of placing the most relevant entries first for the user's convenience. However, to reduce the associated computational overhead, lists are not merged based on an examination of every single entry. Rather, they are merged based on an examination of only a small number of entries from each list. Specifically, there is no requirement for examining the content of each result item."	
'704	1, 12	wherein the representative	Plain and ordinary meaning to a		

		value varies in accordance with predetermined manner / wherein said representative value varies in accordance with said predetermined manner	person of ordinary skill in the art in light of the specification.		
'704	1[1]	predetermined manner	Not indefinite.	'704 Patent at 5:28-43: "FIG. 2 illustrates one embodiment of the processing operations according to the present invention. In typical operation, a query is transmitted to a first search engine, which in turn transmits the query to other search engines (block 70). Eventually, each of these other search engines returns a result list that is received by the first search engine (block 72). The first search engine then begins to merge the result lists according to the processing steps of the present invention. In essence, the result lists are merged with the goal of placing the most relevant entries first for the user's convenience. However, to reduce the associated computational overhead, lists	

[1] In the parties' July 5 meet and confer, Google stated that it believes there may be an antecedent basis issue with the last recitation of this term in claim 1, raising a separate potential dispute than the predetermined manner term already identified.

				are not merged based on an examination of every single entry. Rather, they are merged based on an examination of only a small number of entries from each list. Specifically, there is no requirement for examining the content of each result item." '704 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000343-00000346: "Applicant's particular system and associated methods in the environment of ranking multiple subsets for a result list for a plurality of search engines is the combination of the limitations of selecting a subset of entries from each result list; determining a scoring value for each of the entries of said subsets; characterizing or assigning a representative value to each of said subsets; then merging entries in a predetermined manner in a single list based on said representative value; and wherein the representative value varies in according with said representative value in combination with the other limitations of the claims . . . ."	
'704	1, 12	selecting a subset of entries from each result list	Plain and ordinary meaning to a person of ordinary skill in the art in		Seymor Lipschutz and Marc Lipson, Schaum's Outline of Discrete Mathematics 2 (1997). VALTRUS-GOOGLE-NDTX-00008495-00008506.

			light of the specification.		Orlando A. Oronce and Marilyn O. Mendoza, Exploring Mathematics: Geometry III 11 (2003). VALTRUS-GOOGLE-NDTX-00008507-00008513. E. Kamke, Theory of Sets 6 (1950). VALTRUS-GOOGLE-NDTX-00008478-00008490.
'704	1, 12	producing a merged list of entries / merging entries . . . into a single list	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'704	6, 17	probability value	A value representing how likely a list is to be selected.	'704 Patent at 6:20-22: "Each list is assigned a probability value equal to its representative value's percentage of the total representative values for all lists." '704 Patent at 7:35-54: "According to another embodiment, each list 110, 112, 114 is assigned a probability value equal to its average scoring value's percentage of the total of all average scoring values. Entries are then selected from each list based on its probability value. Here, for instance, the total of all average scoring values 130, 132, 134 is 11.25+14.95+13.225=39.425. This means result list 110 is assigned a probability value equal to (11.25/39.425)100%=28.54%. In like	"Probability." Steven M. Kaplan, Wiley Electrical and Electronics Engineering Dictionary 605 (2004). VALTRUS-GOOGLE-NDTX-00007435-00007455. "Probability." Rudolf F. Graf, Modern Dictionary of Electronics 589 (1999). VALTRUS-GOOGLE-NDTX-00007456-00007470. "Probability." Microsoft Computer Dictionary 423 (Alex Blanton and Sandra Haynes, eds., 5th ed. 2002). VALTRUS-GOOGLE-NDTX-00007483-00007498. "Probability." Dick Pountain, The New Penguin Dictionary of Computing 386

				manner, result list 112 is assigned a probability value of (14.95/39.425)100%=37.92%, and result list 114 is assigned a probability value of (13.225/39.425)100%=33.54%. Result lists are then selected in pseudorandom fashion, where at each selection result list 110 has a 28.54% chance of being picked, list 112 has a 37.92% chance, and list 114 has a 33.54% chance. Once a list is selected, the first entry that has not already been selected is picked. Once every entry in a list is selected, the total of all average scoring values is recalculated without that list's average scoring value, and the process continues until every entry of every list has been selected." '704 Patent at 6:22-28: "Lists are then selected according to their probability value, with lists having higher probability values being more likely to be selected. When a list is selected, the first entry on that list that has not already been selected is picked. This process is repeated, with the total representative value being revised when all entries of a list are picked." '704 Patent Prosecution History, VALTRUS-GOOGLE-NDTX-00000298-00000303: "Applicant's particular system and associated	(2001). VALTRUS-GOOGLE-NDTX-00007517-00007536. "Probability." E.J. Borokowsi and J.M. Borwein, Collins Dictionary of Mathematics 448 (2nd ed. 2007). VALTRUS-GOOGLE-NDTX-00007556-00007567. "Probability." Collins English Dictionary 1291 (7th ed. 2005). VALTRUS-GOOGLE-NDTX-00007568-00007587. "Probability." The New Oxford American Dictionary 1350 (Erin McKean ed., 2nd ed. 2005). VALTRUS-GOOGLE-NDTX-00007588-00007606.

				methods in the environment of ranking multiple result lists for a plurality of search engines is the combination of the limitations of assigning to each result list a probability value based on its average value and the limitation of selecting a result list preferentially based on its probability value so as to form a selected list in combination with the other limitations of the claims, was not disclosed by, would not have been obvious over, nor would have been fairly suggested by the prior art of record or that encountered in searching of the prior art." VALTRUS-GOOGLE-NDTX-00000178-00000187: "Since each source in a schedule succeeds probabilistically, a schedule generates a probability distribution over *outcomes*, where each outcome is one possible way that the schedule's sources might respond to the query." VALTRUS-GOOGLE-NDTX-00000188-00000220. VALTRUS-GOOGLE-NDTX-00000263-00000271.	

II. U.S. Patent No. 6,738,764

Patent	Claim(s)	Term	Valtrus's Proposed Construction	Examples of Intrinsic Evidence	Examples of Extrinsic Evidence
'764	1, 4, 5, 7, 11, 12, 14, 17, 18	relevance score	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'764	1, 4, 5, 7, 11, 12, 14, 17, 18	similarity score	A score that is calculated utilizing a feature vector which characterizes attributes and query words based on a viewed document database, from a Web search engine.	'764 Patent at 1:40-45: "Thus, the order in which the search engine presents the results for a query q may change with time, depending on the behavior of users. Since this technique is time-variant, it is referred to as an adaptive method. In contrast, scoring methods that are time-invariant are referred to as static methods." '764 Patent at 1:65-67: "A similarity score is calculated for the query utilizing a feature vector that characterizes attributes and query words associated with the document." '764 Patent at 2:12-22: "A vector constructor forms a feature vector for each viewed document, each feature vector characterizing attributes associated with a selected viewed	Hans-Peter Kriegel et al., Using Sets of Feature Vectors for Similarity Search on Voxelized CAD Objects (2003). VALTRUS-GOOGLE-NDTX-00007330-00007341. Selim Aksoy and Robert M. Haralick, *Feature normalization and likelihood-based similarity measures for image retrieval*, 22 PATTERN RECOGNITION LETTERS 563, 563 (2001). VALTRUS-GOOGLE-NDTX-00007090-00007109. Dawei Yin et al., Ranking Relevance in Yahoo Search (2016). VALTRUS-GOOGLE-NDTX-00007160-00007169. Hiroshi Shimodaira, Similarity and recommender systems (2015).

				document and query words associated with the selected viewed document. A similarity processor calculates a similarity score for the query utilizing the feature vector of the selected viewed document. A ranking processor assigns a rank value for the selected viewed document based upon a function that incorporates the relevance score and the similarity score for the selected viewed document." '764 Patent at 3:41-42: "As discussed below, the vector characterizes attributes and query words associated with a document." '764 Patent at 3:48-52: "The memory 30 also stores a similarity processor 80. As discussed below, the similarity processor 80 calculates a similarity score between a query and a feature vector of a document. Thus, the similarity processor 80 populates a similarity database 82 with a set of similarity score entries 84." '764 Patent at 4:14-23: "The user may then view a subset of the documents in a given pattern. The viewing and pattern of viewing suggests document relevance. As indicated in FIG. 2, these viewed documents are then logged (step 102). More particularly, the	VALTRUS-GOOGLE-NDTX-00007302-00007306.

				viewed documents, or identifiers of the documents (e.g., pointers), are stored in the viewed document database 50. The entries 52 in the viewed document database 50 may be expressed as $\{(d_j,s_j)_q \mid j \varepsilon [1,n]\}$, which characterizes the subset of viewed documents." '764 Patent at 4:32-44: "A feature vector for a document characterizes attributes and query words associated with a document. The attributes constitute document signatures. Thus, the attributes may be in the form of a list of keywords or other document indicia. Word frequency is often used as the feature value. The query words can also be incorporated into the feature vector. As a result, each document in the collection is augmented by a feature vector, v. This feature vector consists of not only the document signatures but also those query words that might capture information about user's behavior and interest. This process may take a certain period of time in order to build up reliable feature vectors. '764 Patent at 4:53-65: "More particularly, the ranking function adaptively weighs the relevance value and the similarity score based on their quality, as derived from users' behavior.	

				There are different ways to implement the ranking function. One implementation is to apply the ranking function to every document returned by the basic search engine and then re-rank the documents based on a combined scoring function of the relevancy score (from the search engine 42) and the similarity score (from the similarity processor 80). A more sophisticated method is to build an index of the feature vectors, which makes it feasible to compute the similarity score between the query and virtually every document in the collection." '764 Patent at 5:10-6:5: "The key is to adaptively weigh the base score and the similarity score based on the their quality. In one embodiment of the invention the quality measure is derived from users' behavior. Assume that a scoring function or a search engine is good if most clicks are among the top T choices (e.g., corresponding to a page of delivered search results). Let $N_i(Q,T)$ be the total number of viewed documents that appear in the top T candidates for a group of queries Q and N(Q) be the total number of viewed documents for the group of queries Q. The quality of a scoring function is measured by N(Q,	

				T)/N(Q). The larger this value is, the better the quality is. The weights in equation (1) can then be derived from this quality measure as follows. w_i=½$ln[\theta_i/((1-\theta_i], i$=1, 2, where θ_i is a clipped quality measure on s (i=1) or p (i=2) defined as follows. θ_i=MAX(0.5, $N_i(Q, T)/N(Q)$), It is important to point out that θ_i is a function of a group of queries Q. One scoring function can be better than another on a particular set of queries, while another may perform better on a different set of queries. The adaptive weighting scheme of the invention can capture the difference in performance, while a static weighting function cannot. A number of methods can be used to group individual archived queries into query groups. For example, one can assign queries to one of a set of pre-specified categories. All the queries associated with a category belong to a query group. These categories can be defined using the "searching-within-category" constraint associated with search engines. Another approach is to group queries into four groups: (s!, p!),	

				(s, p!), (s!, p), and (s, p), where s indicates that more clicked documents of a query q appear in the top T than outside of the top T, when using relevancy scores. S! is the opposite of s. p and p! have the same definition as s and s!, except that the similarity score p is used. The above scheme can be applied recursively by considering if f(d, s, p, q) were the score of the basic search engine. As more and more feedback is obtained over time, new features will boost the relevant documents to the top T choices by using equation (1) recursively. The scheme requires that feature vectors be indexed periodically. A search engine is preferably scheduled to update the weights in equation (1) daily, weekly, or monthly. Accordingly, users' experience improved performance over the time. The above schemes assume that the similarity measure is pre-defined. As more feedback is obtained over time, one can optimize the similarity measure in such a way that the top T choices of the search results based on the similarity measure will include as many relevant documents that had rank >T (low-rank) by the previous ranking function. A sequence of such similarity measures	

				can be trained, each of which emphasizes the viewed low-rank documents. The final relevant score is then computed as follows. $f(d, s, p, q)=(w_0 s+\Sigma^k_{i=1} w_i p)/(w_0+\Sigma^k_{i=1} w_i)$, It can be shown that the probability of a viewed document being excluded from the top T choices will converge exponentially to zero as k increases, provided that (i) $N_i(Q, T)/N(Q)>0.5$ for all k, and (ii) the number of distinct viewed documents for any query is less than T." '764 Patent at 6:16-20: "A similarity score for the query is then calculated utilizing a feature vector characterizing attributes and query words associated with the document." '764 Patent at 4:45-50: "The next processing step in FIG. 2 is to calculate a similarity score for the query utilizing the feature vector (step 108). Again considering the query q, a similarity measure, p(q,v), can be defined between q and a feature vector v. For example, the commonly used cosine similarity can be used." FIGS. 1 and 2.	

				'764 Patent at 3:41-47: "Each entry 74 includes a document identification value and an associated vector. As discussed below, the vector characterizes attributes and query words associated with a document. The attributes may be key words found within the document. The query words may be query words used in previous searches that resulted in the identification of the document." '764 Patent at 4:30-44: "The contents of the document-query database 62 may then be used to form a feature vector for each viewed document. A feature vector for a document characterizes attributes and query words associated with a document. The attributes constitute document signatures. Thus, the attributes may be in the form of a list of keywords or other document indicia. Word frequency is often used as the feature value. The query words can also be incorporated into the feature vector. As a result, each document in the collection is augmented by a feature vector, v. This feature vector consists of not only the document signatures but also those query words that might capture information about user's behavior and interest. This process may take a certain period of time in order to build up reliable feature vectors."	

				'764 Patent at 1:56-59: "In view of the foregoing, it would be highly desirable to provide a technique that selectively emphasizes a static method or an adaptive method to achieve optimal search results for a given query." '764 Patent at 2:3-21: "The invention also includes a computer readable memory to rank search results. The computer readable memory includes a search engine to produce relevance search results based upon a query, the relevance search results including a list of documents, wherein each document includes an associated relevance score. A viewed document database stores viewed document indicia corresponding to documents viewed in response to the relevance search results. A viewed document processor associates the viewed document indicia with different queries. A vector constructor forms a feature vector for each viewed document, each feature vector characterizing attributes associated with a selected viewed document and query words associated with the selected viewed document. A similarity processor calculates a similarity score for the query utilizing the feature vector of the selected viewed document. A ranking processor assigns a rank value for the selected viewed document based	

				upon a function that incorporates the relevance score and the similarity score for the selected viewed document." '764 Patent at 2:22-29: "The invention also includes a computer readable memory with a search engine to produce a relevance score for a document in view of a query. A similarity processor calculates a similarity score for the query utilizing a feature vector that characterizes attributes and query words associated with the document. A rank processor assigns a rank value to the document based upon the relevance score and the similarity score." '764 Patent Prosecution History, VALTRUS-GOOGLE-NDTX-00000472-00000480: "But nowhere does [prior art] Diamond teach or suggest that a similarity score can be determined using query words of a different query associated with a document, such as a stored document in a database. At most, Diamond discloses determining individual scores, as similarity scores, by comparing a query against subject field codes and terms, both of which appear to be derived solely from the text of a document. . . . As such, the similarity score of Diamond cannot be said to be	

				equivalent to the similarity score of the claimed invention." '764 Patent Prosecution History, VALTRUS-GOOGLE-NDTX-00000483-00000486. '764 Patent Prosecution History, VALTRUS-GOOGLE-NDTX-00000416-00000440. U.S. Patent No. 5,893,095, cited on face of '764 Patent, at 9:56-58: "The output of the image analysis module 122 is a feature vector (FV) that describes the visual object passed to it by one of modules 108, 110 or 112. U.S. Patent No. 5,893,095, cited on face of '764 Patent, at 12:34-36: "A feature vector is a concatenation of a set of feature data elements corresponding to a set of primitives in a schema (further described hereinbelow)." '764 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000428-00000429: "[One] approach to the ranking problem views the set of scores assigned by a ranking function as a vector in a high-dimensional vector space and defines a utility function that takes as input a target vector, a set of vectors that will approximate the target, and some free parameters such as	

				weights to form a linear combination of the set of vectors." U.S. Patent 5,893,095, cited on face of '764 Patent, at Abstract: "A system and method for content-based search and retrieval of visual objects. A base visual information retrieval (VIR) engine utilizes a set of universal primitives to operate on the visual objects. An extensible VIR engine allows custom, modular primitives to be defined and registered. A custom primitive addresses domain specific problems and can utilize any image understanding technique. Object attributes can be extracted over the entire image or over only a portion of the object. A schema is defined as a specific collection of primitives. A specific schema implies a specific set of visual features to be processed and a corresponding feature vector to be used for content-based similarity scoring. A primitive registration interface registers custom primitives and facilitates storing of an analysis function and a comparison function to a schema table. A heterogeneous comparison allows objects analyzed by different schemas to be compared if at least one primitive is in common between the schemas. A threshold-based comparison is utilized to improve performance of the VIR	

				engine. A distance between two feature vectors is computed in any of the comparison processes so as to generate a similarity score." U.S. Patent 5,893,095, cited on face of '764 Patent, at 7:42-67: "One presently preferred implementation is as follows. For visual information, features may belong to five abstract data types: values, distributions, indexed values, indexed distributions, and graphs. A value is, in the general case, a set of vectors that may represent some global property of the image. The global color of an image, for example, can be a vector of RGB values, while the dominant colors of an image can be defined as the set of k most frequent RGB vectors in an image. A distribution, such as a color histogram is typically defined on an n-dimensional space which has been partitioned into b buckets. Thus, it is a b-dimensional vector. An indexed value is a value local to a region of an image or a time point in a video or both; as a data type it is an indexed set of vectors. The index can be one-dimensional as in the key-frame number for a video, or it can be multi-dimensional as in the orthonormal bounding box coordinates covering an image segment. An indexed distribution is a local pattern such as the intensity	

				profile of a region of interest, and can be derived from a collection of b-dimensional vectors by introducing an index. A graph represents relational information, such as the relative spatial position of two regions of interest in an image. We do not consider a graph as a primary type of interest, because it can be implemented in terms of the other four data types, with some application-dependent rules of interpretation (e.g. transitivity of spatial predicates, such as left-of)." U.S. Patent 5,893,095, cited on face of '764 Patent, at 3:15-52: "A typical content-based retrieval system might be described as follows: image features are precomputed during an image insertion phase. These representations may include characteristics such as local intensity histograms, edge histograms, region-based moments, spectral characteristics, and so forth. These features are then stored in a database as structured data. A typical query involves finding the images which are "visually similar" to a given candidate image. In order to submit a query, a user presents (or constructs) a candidate image. This query image may already have features associated with it (i.e., an image which already exists within the database), or may be novel, in which	

				case a characterization is performed "on the fly" to generate features. Once the query image has been characterized, the query executes by comparing the features of the candidate image against those of other images in the database. The result of each comparison is a scalar score which indicates the degree of similarity. This score is then used to rank order the results of the query. This process can be extremely fast because image features are pre-computed during the insertion phase, and distance functions have been designed to be extremely efficient at query time. There are many variants on this general scheme, such as allowing the user to express queries directly at the feature level, combining images to form queries, querying over regions of interest, and so forth. General systems (using color, shape, etc.) are adequate for applications with a broad image domain, such as generic stock photography. In general, however, these systems are not applicable to specific, constrained domains. It is not expected, for example, that a texture similarity measure that works well for nature photography will work equally well for mammography. If mammogram databases need to be searched by image content, one would need to develop	

				specific features and similarity measures. This implies that a viable content-based image retrieval system will have to provide a mechanism to define arbitrary image domains and allow a user to query on a user-defined schema of image features and similarity metrics." U.S. Patent 5,893,095, cited on face of '764 Patent, at 4:4-13: "Some of these features are computed globally, i.e., over an entire image, and some are local, i.e., computed over a small region in the image. The VIR Engine expresses visual features as image "primitives". Primitives can be very general (such as color, shape, or texture) or quite domain specific (face recognition, cancer cell detection, etc.). The basic philosophy underlying this architecture is a transformation from the data-rich representation of explicit image pixels to a compact, semantic-rich representation of visually salient characteristics." U.S. Patent 5,963,940, cited on face of '764 Patent, at 9:51-61: "Each information bearing word in a text is looked up in the online, lexical resource. If the word is in the lexicon, it is assigned a single, unambiguous subject code using, if necessary, a	

				process of disambiguation. Once each content-bearing word in a text has been assigned a single SFC, the frequencies of the codes for all words in the document are combined to produce a fixed length, subject-based vector representation of the document's contents. This relatively high-level, conceptual representation of documents and queries is an important representation of texts used for later matching and ranking." U.S. Patent 6,026,388, cited on face of '764 Patent, at 10:1-11: "Each information bearing word in a text is looked up in the online, lexical resource. If the word is in the lexicon, it is assigned a single, unambiguous subject code using, if necessary, a process of disambiguation. Once each content-bearing word in a text has been assigned a single SFC, the frequencies of the codes for all words in the document are combined to produce a fixed length, subject-based vector representation of the document's contents. This relatively high-level, conceptual representation of documents and queries is an important representation of texts used for later matching and ranking."	

				U.S. Patent 6,289,353, cited on face of '764 Patent, at 1:46-59: "On the other hand, some large-scale systems which lack mechanisms for adaptation have successfully exploited the statistical relationships among, documents and terms found in those documents, for storage and retrieval of documents and other information items. For example, U.S. Pat. No. 5,619,709 to Caid, et. al., describes generation of context vectors that represent conceptual relationships among information items. The context vectors in Caid, et. al. are developed based on word proximity in a static training corpus. The context vectors do not adapt to user profile information, new information sources, or user feedback regarding the relevancy of documents retrieved by the system. Thus, the system in Caid, et. al. does not evolve over time to provide more relevant document retrieval." U.S. Patent 6,289,353, cited on face of '764 Patent, at 2:59-3:9: "Machine learning techniques are used to facilitate automated emergence of useful mathematical spaces in which information elements are represented as vectors of real numbers. A first machine learning technique automatically generates a set of axes that characterize the central semantic dimensions of a	

				collator's set of documents. The procedure begins with the set of documents coded as vectors of term frequencies in an information space spanned by a dictionary of all terms in the set. The collator then finds a reduced dimensionality space spanned by a set of concepts which are central to a significant portion of the set of documents. The original information space, spanned by the entire dictionary, is mapped into a low-dimensional space spanned by a set of central concepts. The new low-dimensional space represents a particular view of the portion of the database represented by the collator's set of documents. The database portion is not chosen in advance, but evolves contemporaneously with the vector space structure which emerges." U.S. Patent 5,642,502, cited on face of '764 Patent, at Abstract: "Search system and method for retrieving relevant documents from a text data base collection comprised of patents, medical and legal documents, journals, news stories and the like. Each small piece of text within the documents such as a sentence, phrase and semantic unit in the data base is treated as a document. Natural language queries are used to search for relevant documents from the	

				data base. A first search query creates a selected group of documents. Each word in both the search query and in the documents are given weighted values. Combining the weighted values creates similarity values for each document which are then ranked according to their relevant importance to the search query. A user reading and passing through this ranked list checks off which documents are relevant or not. Then the system automatically causes the original search query to be updated into a second search query which can include the same words, less words or different words than the first search query. Words in the second search query can have the same or different weights compared to the first search query. The system automatically searches the text data base and creates a second group of documents, which as a minimum does not include at least one of the documents found in the first group. The second group can also be comprised of additional documents not found in the first group. The ranking of documents in the second group is different than the first ranking such that the more relevant documents are found closer to the top of the list." U.S. Patent 5,893,095, cited on face of '764 Patent, at 4:39-57: "An important	

				concept in content-based retrieval is to determine how similar two pictures are to one another. The notion of similarity (versus exact matching as in database systems) is appropriate for visual information because multiple pictures of the same scene will not necessarily "match," although they are identical in content. In the paradigm of content-based retrieval, pictures are not simply matched, but are ranked in order of their similarity to the query image. Another benefit is that content extraction results in very high information compression. The content of an image file may be expressed in as little as several hundred bytes of memory, regardless of the original image size. As an image is inserted into a VIMSYS database, the system extracts the content in terms of generic image properties such as its color, texture, shape and composition, and uses this information for all subsequent database operations. Except for display, the original image is not accessed. Naturally, the VIMSYS model also supports textual attributes as do all standard databases." U.S. Patent 5,642,502, cited on face of '764 Patent, at 1:35-42: "Statistically based text retrieval systems generally rank retrieved documents according to their statistical similarity to a user's	

				search request(referred to often as the query). Statistically based systems provide advantages over traditional Boolean retrieval methods, especially for users of such systems, mainly because they allow for natural language input." U.S. Patent 5,642,502, cited on face of '764 Patent, at FIG. 2.	
'764	1, 2, 3, 7, 9, 10, 14, 15, 16	feature vector [cl. 1: that characterizes attributes and query words of a different query associated with said document]	A vector, which characterizes attributes and query words based on a viewed document database, from a Web search engine.	'764 Patent at 3:41-47: "Each entry 74 includes a document identification value and an associated vector. As discussed below, the vector characterizes attributes and query words associated with a document. The attributes may be key words found within the document. The query words may be query words used in previous searches that resulted in the identification of the document." '764 Patent at 1:65-66: "A similarity score is calculated for the query utilizing a feature vector that characterizes attributes and query words associated with the document." '764 Patent at 4:30-44: "The contents of the document-query database 62 may then be used to form a feature vector for each viewed document. A feature vector for a document characterizes attributes	"Feature." Dictionary of Computer Science, Engineering, and Technology 180 (Phillip A. Laplante ed., 2nd ed. 2001). VALTRUS-GOOGLE-NDTX-00007416-00007434. "Feature." Steven M. Kaplan, Wiley Electrical and Electronics Engineering Dictionary 277 (2004). VALTRUS-GOOGLE-NDTX-00007435-00007455. "Feature." Microsoft Computer Dictionary 208 (Alex Blanton and Sandra Haynes, eds., 5th ed. 2002). VALTRUS-GOOGLE-NDTX-00007483-00007498. "Feature." Harry Newton, Newton's Telecom Dictionary 390-91 (24th ed. 2008). VALTRUS-GOOGLE-NDTX-00007499-00007516.

				and query words associated with a document. The attributes constitute document signatures. Thus, the attributes may be in the form of a list of keywords or other document indicia. Word frequency is often used as the feature value. The query words can also be incorporated into the feature vector. As a result, each document in the collection is augmented by a feature vector, v. This feature vector consists of not only the document signatures but also those query words that might capture information about user's behavior and interest. This process may take a certain period of time in order to build up reliable feature vectors." '764 Patent at 4:14-23: "The user may then view a subset of the documents in a given pattern. The viewing and pattern of viewing suggests document relevance. As indicated in FIG. 2, these viewed documents are then logged (step 102). More particularly, the viewed documents, or identifiers of the documents (e.g., pointers), are stored in the viewed document database 50. The entries 52 in the viewed document database 50 may be expressed as $\{(d_j,s_j)_q \| j\varepsilon[1,n]\}$, which characterizes the subset of viewed documents."	"Feature." Collins English Dictionary 595 (7th ed. 2005). VALTRUS-GOOGLE-NDTX-00007568-00007587. "Feature." The New Oxford American Dictionary 615 (Erin McKean ed., 2nd ed. 2005). VALTRUS-GOOGLE-NDTX-00007588-00007606. "Vector." Alan Freedman, Computer Desktop Encyclopedia 518 (9th ed. 2001). VALTRUS-GOOGLE-NDTX-00007401-00007415. "Vector." Dictionary of Computer Science, Engineering, and Technology 180 (Phillip A. Laplante ed., 2nd ed. 2001). VALTRUS-GOOGLE-NDTX-00007416-00007434. "Vector." Steven M. Kaplan, Wiley Electrical and Electronics Engineering Dictionary 833 (2004). VALTRUS-GOOGLE-NDTX-00007435-00007455. "Vector." Rudolf F. Graf, Modern Dictionary of Electronics 826 (1999). VALTRUS-GOOGLE-NDTX-00007456-00007470. "Vector." S.M.H. Collin, Dictionary of Computing 348 (5th ed. 2004).

				FIG. 2. '764 Patent at 1:56-59: "In view of the foregoing, it would be highly desirable to provide a technique that selectively emphasizes a static method or an adaptive method to achieve optimal search results for a given query." '764 Patent at 2:3-21: "The invention also includes a computer readable memory to rank search results. The computer readable memory includes a search engine to produce relevance search results based upon a query, the relevance search results including a list of documents, wherein each document includes an associated relevance score. A viewed document database stores viewed document indicia corresponding to documents viewed in response to the relevance search results. A viewed document processor associates the viewed document indicia with different queries. A vector constructor forms a feature vector for each viewed document, each feature vector characterizing attributes associated with a selected viewed document and query words associated with the selected viewed document. A similarity processor calculates a similarity score for the query utilizing the feature vector of the selected viewed document. A	VALTRUS-GOOGLE-NDTX-00007471-00007482. "Vector." Microsoft Computer Dictionary 548 (Alex Blanton and Sandra Haynes, eds., 5th ed. 2002). VALTRUS-GOOGLE-NDTX-00007483-00007498. "Vector." Harry Newton, Newton's Telecom Dictionary 986 (24th ed. 2008). VALTRUS-GOOGLE-NDTX-00007499-00007516. "Vector." Dick Pountain, The New Penguin Dictionary of Computing 526-27 (2001). VALTRUS-GOOGLE-NDTX-00007517-00007536. "Vector." Collins English Dictionary 1777 (7th ed. 2005). VALTRUS-GOOGLE-NDTX-00007568-00007587. "Vector." E.J. Borokowsi and J.M. Borwein, Collins Dictionary of Mathematics 595 (2nd ed. 2007). VALTRUS-GOOGLE-NDTX-00007556-00007567. "Vector." The New Oxford American Dictionary 1863 (Erin McKean ed., 2nd ed. 2005). VALTRUS-GOOGLE-NDTX-00007588-00007606.

				ranking processor assigns a rank value for the selected viewed document based upon a function that incorporates the relevance score and the similarity score for the selected viewed document." '764 Patent at 2:22-29: "The invention also includes a computer readable memory with a search engine to produce a relevance score for a document in view of a query. A similarity processor calculates a similarity score for the query utilizing a feature vector that characterizes attributes and query words associated with the document. A rank processor assigns a rank value to the document based upon the relevance score and the similarity score." '764 Patent at 5:10-6:5: "The key is to adaptively weigh the base score and the similarity score based on the their quality. In one embodiment of the invention the quality measure is derived from users' behavior. Assume that a scoring function or a search engine is good if most clicks are among the top T choices (e.g., corresponding to a page of delivered search results). Let $N_i(Q,T)$ be the total number of viewed documents that appear in the top T candidates for a	"Feature Vector." William Raynor, The International Dictionary of Artificial Intelligence 106 (1999). VALTRUS-GOOGLE-NDTX-00007607-00007901. Sergey Brin and Lawrence Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine (1998). VALTRUS-GOOGLE-NDTX-00007266-00007285. Hans-Peter Kriegel et al., Using Sets of Feature Vectors for Similarity Search on Voxelized CAD Objects (2003). VALTRUS-GOOGLE-NDTX-00007330-00007341. Selim Aksoy and Robert M. Haralick, *Feature normalization and likelihood-based similarity measures for image retrieval*, 22 PATTERN RECOGNITION LETTERS 563, 563 (2001). VALTRUS-GOOGLE-NDTX-00007090-00007109. Dawei Yin et al., Ranking Relevance in Yahoo Search (2016). VALTRUS-GOOGLE-NDTX-00007160-00007169. Hiroshi Shimodaira, Similarity and recommender systems (2015). VALTRUS-GOOGLE-NDTX-00007302-00007306.

				group of queries Q and N(Q) be the total number of viewed documents for the group of queries Q. The quality of a scoring function is measured by N(Q, T)/N(Q). The larger this value is, the better the quality is. The weights in equation (1) can then be derived from this quality measure as follows. $w_i = \frac{1}{2} ln[\theta_i/((1-\theta_i], i=1, 2,$ where θ_i is a clipped quality measure on s (i=1) or p (i=2) defined as follows. $\theta_i = MAX(0.5, N_i(Q, T)/N(Q)),$ It is important to point out that θ_i is a function of a group of queries Q. One scoring function can be better than another on a particular set of queries, while another may perform better on a different set of queries. The adaptive weighting scheme of the invention can capture the difference in performance, while a static weighting function cannot. A number of methods can be used to group individual archived queries into query groups. For example, one can assign queries to one of a set of pre-specified categories. All the queries associated with a category belong to a query group. These categories can be	

				defined using the "searching-within-category" constraint associated with search engines. Another approach is to group queries into four groups: (s!, p!), (s, p!), (s!, p), and (s, p), where s indicates that more clicked documents of a query q appear in the top T than outside of the top T, when using relevancy scores. S! is the opposite of s. p and p! have the same definition as s and s!, except that the similarity score p is used. The above scheme can be applied recursively by considering if f(d, s, p, q) were the score of the basic search engine. As more and more feedback is obtained over time, new features will boost the relevant documents to the top T choices by using equation (1) recursively. The scheme requires that feature vectors be indexed periodically. A search engine is preferably scheduled to update the weights in equation (1) daily, weekly, or monthly. Accordingly, users' experience improved performance over the time. The above schemes assume that the similarity measure is pre-defined. As more feedback is obtained over time, one can optimize the similarity measure in such a way that the top T choices of the search results based on the similarity	

				measure will include as many relevant documents that had rank >T (low-rank) by the previous ranking function. A sequence of such similarity measures can be trained, each of which emphasizes the viewed low-rank documents. The final relevant score is then computed as follows. $f(d, s, p, q)=(w_0 s+\Sigma^k_{i=1} w_i p)/(w_0+\Sigma^k_{i=1} w_i)$, It can be shown that the probability of a viewed document being excluded from the top T choices will converge exponentially to zero as k increases, provided that (i) $N_i(Q, T)/N(Q)>0.5$ for all k, and (ii) the number of distinct viewed documents for any query is less than T." '764 Patent at 6:16-20: "A similarity score for the query is then calculated utilizing a feature vector characterizing attributes and query words associated with the document." FIGS. 1, 2, and 3. U.S. Patent No. 5,893,095, cited on face of '764 Patent, at 9:56-58: "The output of the image analysis module 122 is a feature vector (FV) that describes the	

				visual object passed to it by one of modules 108, 110 or 112. U.S. Patent No. 5,893,095, cited on face of '764 Patent, at 12:34-36: "A feature vector is a concatenation of a set of feature data elements corresponding to a set of primitives in a schema (further described hereinbelow)." '764 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000428-00000429: "[One] approach to the ranking problem views the set of scores assigned by a ranking function as a vector in a high-dimensional vector space and defines a utility function that takes as input a target vector, a set of vectors that will approximate the target, and some free parameters such as weights to form a linear combination of the set of vectors." U.S. Patent 5,893,095, cited on face of '764 Patent, at Abstract: "A system and method for content-based search and retrieval of visual objects. A base visual information retrieval (VIR) engine utilizes a set of universal primitives to operate on the visual objects. An extensible VIR engine allows custom, modular primitives to be defined and registered. A custom primitive addresses domain specific problems and	

				can utilize any image understanding technique. Object attributes can be extracted over the entire image or over only a portion of the object. A schema is defined as a specific collection of primitives. A specific schema implies a specific set of visual features to be processed and a corresponding feature vector to be used for content-based similarity scoring. A primitive registration interface registers custom primitives and facilitates storing of an analysis function and a comparison function to a schema table. A heterogeneous comparison allows objects analyzed by different schemas to be compared if at least one primitive is in common between the schemas. A threshold-based comparison is utilized to improve performance of the VIR engine. A distance between two feature vectors is computed in any of the comparison processes so as to generate a similarity score." U.S. Patent 5,893,095, cited on face of '764 Patent, at 7:42-67: "One presently preferred implementation is as follows. For visual information, features may belong to five abstract data types: values, distributions, indexed values, indexed distributions, and graphs. A value is, in the general case, a set of vectors that may represent some global	

				property of the image. The global color of an image, for example, can be a vector of RGB values, while the dominant colors of an image can be defined as the set of k most frequent RGB vectors in an image. A distribution, such as a color histogram is typically defined on an n-dimensional space which has been partitioned into b buckets. Thus, it is a b-dimensional vector. An indexed value is a value local to a region of an image or a time point in a video or both; as a data type it is an indexed set of vectors. The index can be one-dimensional as in the key-frame number for a video, or it can be multi-dimensional as in the orthonormal bounding box coordinates covering an image segment. An indexed distribution is a local pattern such as the intensity profile of a region of interest, and can be derived from a collection of b-dimensional vectors by introducing an index. A graph represents relational information, such as the relative spatial position of two regions of interest in an image. We do not consider a graph as a primary type of interest, because it can be implemented in terms of the other four data types, with some application-dependent rules of interpretation (e.g. transitivity of spatial predicates, such as left-of)."	

				U.S. Patent 5,893,095, cited on face of '764 Patent, at 3:15-52: "A typical content-based retrieval system might be described as follows: image features are precomputed during an image insertion phase. These representations may include characteristics such as local intensity histograms, edge histograms, region-based moments, spectral characteristics, and so forth. These features are then stored in a database as structured data. A typical query involves finding the images which are "visually similar" to a given candidate image. In order to submit a query, a user presents (or constructs) a candidate image. This query image may already have features associated with it (i.e., an image which already exists within the database), or may be novel, in which case a characterization is performed "on the fly" to generate features. Once the query image has been characterized, the query executes by comparing the features of the candidate image against those of other images in the database. The result of each comparison is a scalar score which indicates the degree of similarity. This score is then used to rank order the results of the query. This process can be extremely fast because image features are pre-computed during the insertion phase, and distance functions have been designed to be	

				extremely efficient at query time. There are many variants on this general scheme, such as allowing the user to express queries directly at the feature level, combining images to form queries, querying over regions of interest, and so forth. General systems (using color, shape, etc.) are adequate for applications with a broad image domain, such as generic stock photography. In general, however, these systems are not applicable to specific, constrained domains. It is not expected, for example, that a texture similarity measure that works well for nature photography will work equally well for mammography. If mammogram databases need to be searched by image content, one would need to develop specific features and similarity measures. This implies that a viable content-based image retrieval system will have to provide a mechanism to define arbitrary image domains and allow a user to query on a user-defined schema of image features and similarity metrics." U.S. Patent 5,893,095, cited on face of '764 Patent, at 4:4-13: "Some of these features are computed globally, i.e., over an entire image, and some are local, i.e., computed over a small region	

				in the image. The VIR Engine expresses visual features as image "primitives". Primitives can be very general (such as color, shape, or texture) or quite domain specific (face recognition, cancer cell detection, etc.). The basic philosophy underlying this architecture is a transformation from the data-rich representation of explicit image pixels to a compact, semantic-rich representation of visually salient characteristics." U.S. Patent 5,963,940, cited on face of '764 Patent, at 9:51-61: "Each information bearing word in a text is looked up in the online, lexical resource. If the word is in the lexicon, it is assigned a single, unambiguous subject code using, if necessary, a process of disambiguation. Once each content-bearing word in a text has been assigned a single SFC, the frequencies of the codes for all words in the document are combined to produce a fixed length, subject-based vector representation of the document's contents. This relatively high-level, conceptual representation of documents and queries is an important representation of texts used for later matching and ranking."	

				U.S. Patent 6,026,388, cited on face of '764 Patent, at 10:1-11: "Each information bearing word in a text is looked up in the online, lexical resource. If the word is in the lexicon, it is assigned a single, unambiguous subject code using, if necessary, a process of disambiguation. Once each content-bearing word in a text has been assigned a single SFC, the frequencies of the codes for all words in the document are combined to produce a fixed length, subject-based vector representation of the document's contents. This relatively high-level, conceptual representation of documents and queries is an important representation of texts used for later matching and ranking." U.S. Patent 6,289,353, cited on face of '764 Patent, at 1:46-59: "On the other hand, some large-scale systems which lack mechanisms for adaptation have successfully exploited the statistical relationships among, documents and terms found in those documents, for storage and retrieval of documents and other information items. For example, U.S. Pat. No. 5,619,709 to Caid, et. al., describes generation of context vectors that represent conceptual relationships among information items. The context vectors in Caid, et. al. are developed	

				based on word proximity in a static training corpus. The context vectors do not adapt to user profile information, new information sources, or user feedback regarding the relevancy of documents retrieved by the system. Thus, the system in Caid, et. al. does not evolve over time to provide more relevant document retrieval." U.S. Patent 6,289,353, cited on face of '764 Patent, at 2:59-3:9: "Machine learning techniques are used to facilitate automated emergence of useful mathematical spaces in which information elements are represented as vectors of real numbers. A first machine learning technique automatically generates a set of axes that characterize the central semantic dimensions of a collator's set of documents. The procedure begins with the set of documents coded as vectors of term frequencies in an information space spanned by a dictionary of all terms in the set. The collator then finds a reduced dimensionality space spanned by a set of concepts which are central to a significant portion of the set of documents. The original information space, spanned by the entire dictionary, is mapped into a low-dimensional space spanned by a set of central concepts. The new low-dimensional space	

				represents a particular view of the portion of the database represented by the collator's set of documents. The database portion is not chosen in advance, but evolves contemporaneously with the vector space structure which emerges."	
'764	1, 4, 5, 6, 7, 11, 12, 13, 14, 17, 18, 19	rank value	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'764	1, 14	in view of a query	A query is one or more words that cause a search engine to return documents it deems relevant	'764 Patent at 1:14-18: "A text search engine receives from a user one or more words of text that form a query. The query may include other search operators, such as Boolean operators, proximity operators and the like. The search engine returns documents that it deems relevant to the query." '764 Patent at 1:36-45: "On receiving query q, the search engine first retrieves all documents that match the query q; it then ranks them in decreasing order of the values N(d,q). This technique is described in U.S. Pat. Nos. 6,006,222 and 6,014,665. Thus, the order in which the search engine presents the results for a query q may change with time, depending on the behavior of users.	"Query." Alan Freedman, Computer Desktop Encyclopedia 816-17 (9th ed. 2001). VALTRUS-GOOGLE-NDTX-00007401-00007415. "Query." Dictionary of Computer Science, Engineering, and Technology 399 (Phillip A. Laplante ed., 2nd ed. 2001). VALTRUS-GOOGLE-NDTX-00007416-00007434. "Query." Steven M. Kaplan, Wiley Electrical and Electronics Engineering Dictionary 621 (2004). VALTRUS-GOOGLE-NDTX-00007435-00007455. "Query." Microsoft Computer Dictionary 433 (Alex Blanton and Sandra Haynes,

				Since this technique is time-variant, it is referred to as an adaptive method. In contrast, scoring methods that are time-invariant are referred to as static methods." '764 Patent at 3:1-5: "Based upon the query, the search engine 42 produces a set of relevance search results 44, which include individual entries 46. The individual entries 46 typically include a document identification and an associated relevance score." '764 Patent at 4:4-8: "The first operation illustrated in FIG. 2 is to produce relevance search results based upon a query (step 100). As previously indicated, a standard search engine 42 may be used to process a query 40 and generate relevance search results 44." '764 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000424-00000426: "One of the primary tasks of information retrieval is searching a large collection of documents for those relevant to a particular query . . . ." '764 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000428-00000430: "Each ranking function, for example a keyword search,	eds., 5th ed. 2002). VALTRUS-GOOGLE-NDTX-00007483-00007498. "Query." Harry Newton, Newton's Telecom Dictionary 761 (24th ed. 2008). VALTRUS-GOOGLE-NDTX-00007499-00007516. "Query." Dick Pountain, The New Penguin Dictionary of Computing 400 (2001). VALTRUS-GOOGLE-NDTX-00007517-00007536. "Query." Collins English Dictionary 1327 (7th ed. 2005). VALTRUS-GOOGLE-NDTX-00007568-00007587. "Query." The New Oxford American Dictionary 1388 (Erin McKean ed., 2nd ed. 2005). VALTRUS-GOOGLE-NDTX-00007588-00007606.

				orders the documents it returns according to their predicted relevance to the query." '764 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000472-00000480: "For example, Diamond neither teaches nor suggests using query words of a different query associated with a document to calculate a similarity score of the claimed invention."	
'764	1, 2, 7, 9, 14, 15	attribute(s)	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'764	1, 7, 14	query words of a different query associated with said document / query words of said different queries associated with said selected viewed document	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		

'764	3, 10, 16[2]	the previous identification of said document	Not indefinite.		
'764	6, 13, 19	linear combination function	a weighted sum of scores	'764 Patent at 5:4-10: "One embodiment of the invention utilizes a linear combination scheme for simplicity: $f(d, s, p, q)=(w_1 s+w_2 p)/(w_1 +w_2)$, where w_1 and w_2 are weights for the base score s and similarity score p, respectively." '764 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000428-00000430: "Such an approach to the ranking problem views the set of scores assigned by a ranking function as a vector in a high-dimensional vector space and defines a utility function that takes as input a target vector, a set of vectors that will approximate the target, and some free parameters such as weights to form a linear combination of the set of vectors." U.S. Patent 5,893,095, cited on face of '764 Patent, at 13:5-10: "The final	"Linear Combination." E.J. Borokowsi and J.M. Borwein, Collins Dictionary of Mathematics 329 (2nd ed. 2007). VALTRUS-GOOGLE-NDTX-00007556-00007567. Joseph Rabinoff, Systems of Linear Equations: Geometry, at Slide 10. VALTRUS-GOOGLE-NDTX-00007110-00007131. Jeffrey Wang, NoBS Linear Algebra, at Slide 38 (May 1, 2018). VALTRUS-GOOGLE-NDTX-00007000-00007089. Digital Image Processing, Digital Image Fundamentals – II, at Slide 12 (June 12, 2017). VALTRUS-GOOGLE-NDTX-00007237-00007265.

[2] In the parties' July 5 meet and confer, Google stated that it believes there may be an antecedent basis issue with the recitations of this term.

				combined score may, for instance, be generated by a linear combination or a weighted sum as follows: $s_f = \sum_i w_i s_i$ " U.S. Patent 6,269,368, cited on face of '764 Patent, at 16:39-18:33: "4.3 Implementing the Score Combiner using a Neural Network and Linear Functions In a preferred embodiment, linear combination functions are used to define the dynamic combination regime. Based on the input information, weight determinator **55**WD determines linear coefficients or weights to be applied to each of the individual match scores. In a specific embodiment, weight determinator **55**WD is implemented as a neural network that receives the input information and adjusts linear coefficients for each of the alternative representation match scores. In a specific embodiment, a feed-forward, multi-layer neural network is used whose output nodes are the linear weights to be applied to each of the individual match scores. The neural network models functions using a set of nodes arranged into layers including an input layer, and output layer, and one or	

				more intermediate hidden layers. The nodes are connected to each other by weighted links. FIG. 6 depicts an example of a neural network which may be used to implement weight determinator **55**WD. As shown in FIG. 6, the neural network comprises a set of nodes arranged into layers, including an input layer, an output layer, and a hidden layer. The nodes are connected to each other by weighted links. The first layer of nodes is made up of a number "n" of input nodes labeled ($N_{1,1}$), ($N_{1,2}$), . . . , ($N_{1,n}$) corresponding to the number of input features which may include query specific features, document specific features, score correlation features, and optionally relevance feedback information. The second layer of the neural network comprises "m" number of hidden nodes labeled ($N_{2,1}$), ($N_{2,2}$), . . . , ($N_{2,m}$). The third layer of the neural network comprises output nodes corresponding to weights to be applied to individual scores generated by the individual matchers **55** *a*-**55** *c*. While FIG. 6 shows five output nodes labeled ($N_{3,1}$), ($N_{3,2}$), ($N_{3,3}$), ($N_{3,4}$), and ($N_{3,5}$) for simplicity, the specific embodiment of the invention uses six output nodes, corresponding to six different alternative representations for documents and queries. The node layers	

				are connected by links characterized by numeric weights <$W_{l,i,j}$> which connect the i node in layer "l" to the jth node in. layer (l+1). Each node of the input layer (Layer 1) has a link to each of the hidden nodes (Layer **2**), and each of the hidden nodes has a link to each of the output nodes (Layer 3). When an input (x_1, . . . , x_n) is presented to the network, the input is propagated towards the output layer in the following manner: (1) Node ($N_{1,1}$) sends its input (x_1) to node ($N_{2,1}$) which receives it as (x_1*$W_{1,1,1}$). Similarly, node ($N_{1,1}$) sends its input x_1 to the remaining nodes in the hidden layer, which receive it as the product of x_1 and the weight associated with the link between ($N_{1,1}$) and the node in the hidden layer. Node ($N_{1,2}$) sends its input (x_2) to node ($N_{2,1}$) which receives it as ($x_{2*W\ 1,2,1}$). Similarly, node ($N_{1,2}$) sends input x_2 to the remaining nodes in the hidden layer, which receive it as the product of x_2 and the weight associated with the link between ($N_{2,1}$) and the node in the hidden layer. The remaining inputs nodes propagate their values to the hidden layer in the same fashion. (2) Node ($N_{2,1}$) sums the signals it receives (x_1*$W_{1,1,1}$+ . . . +x_1*$W_{1,n,1}$), and applies a sigmoid function, to generate its output, ($O_{2,1}$), which is then sent to	

				node ($N_{3,1}$) as ($O_{2,1}$*$W_{2,1,1}$), to node ($N_{3,2}$) as ($O_{2,1}$*$W_{2,1,2}$), and similarly to nodes ($N_{3,3}$) to ($N_{3,5}$) as ($O_{2,1}$*$W_{2,1,3}$), ($O_{2,1}$*$W_{1,2,4}$), and ($O_{2,1}$*$W_{2,1,5}$), respectively. Nodes ($N_{2,2}$) to ($N_{2,m}$) perform similar operations on their received signals to produce output signals ($O_{2,i}$*$W_{2,i,j}$), where "i" is the node number in the hidden layer, and "j" is the node number in the output layer. The sigmoid function "squashes" an input by imposing upper and lower asymptotes on the output as the input goes to positive or negative infinity. A common sigmoid is the logistic transformation: f □ (x) = 1 1 + e - x . $f(x) = \frac{1}{1 + e^{-x}}.$ (3) Node ($N_{3,1}$) sums the signals it receives ($O_{2,1}$*$W_{2,1,1}$+ . . . +$O_{2,m}$*$W_{2,m,1}$), and applies a sigmoid function, to generate its output, (y_1). Nodes ($N_{3,2}$) to ($N_{3,5}$) perform similar operations on the signals they receive to generate outputs (y_2) to (y_5). Although only five outputs are shown in FIG. 6, the specific embodiment of the invention generates six outputs corresponding to the six different alternative representations for documents and queries. An important characteristic of the technique described above is that it is capable of approximating any function,	

				given a sufficient number of hidden layers and nodes [Hertz91 pp. 142-4]. The internal weights for links are set by "training" the neural network. Feedforward networks are generally trained using a technique called "backpropagation." In backpropagation, the difference between a training case's "target" output and its "actual" output is propagated back towards the input nodes in the form of weight adjustments. The formulae used for the weight adjustments depend on an error function applied to this difference, usually the square of the difference. Each training case is presented to the network for weight adjustment, until the training cases have been exhausted. Then another iteration of the same training cases is presented, possibly in a different order. Iterations continue until the sum of squared errors across all training cases falls below some threshold, or until a maximum number of iterations is reached. To prevent overfitting, the weights attained after each iteration are applied to a test sample of cases that are different from the training cases. If the sum of squared errors for the test sample for an iteration exceeds the sum for the previous iteration, the network may have begun to represent "noise" in its weights, indicating that training should be	

				stopped. This protection against overfitting is referred to as cross-validation. The present invention uses a variant of the general backpropagation algorithm since the usual backpropagation algorithm is not appropriate for training the neural network to predict the optimal weights for the linear fusion functions. The goal of the training is to predict a set of outputs that, when used in the linear fusion function, will result in the highest possible ranking of relevant documents. To optimize the network link weights for this goal, the error that is propagated back through the network is inflated by a function of the difference between the ranking of the relevant documents (for the current set of network link weights) and the best ranking possible. The training technique is thus designed to predict the linear weights so as to maximize the proportion of relevant documents to retrieved documents. It should be apparent to those of ordinary skill in the art that although a specific neural network implementation is described above, other neural network implementations and training procedures are also encompassed within the scope of the present invention."	

'764	7	viewed document database	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'764	7, 9, 10, 15, 16	vector constructor	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'764	7	associated with said selected viewed document	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'764	7	similarity processor	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		

III. U.S. Patent No. 7,346,604

Patent	Claim(s)	Term	Valtrus's Proposed Construction	Examples of Intrinsic Evidence	Examples of Extrinsic Evidence
'604	1, 5, 6, 7, 11, 12, 13, 14, 15, 16, 19, 20	expert document(s)	A document that is about one or certain topics and has links to many non-affiliated documents on those topics.	'604 Patent at 2:54-56: "An 'expert page' is a page that is about a certain topic and has links to many 'non-affiliated' pages on that topic." '604 Patent at 4:48-64: "An expert page is a page that is about a certain topic and has links to many non-affiliated pages on that topic. Two pages are non-affiliated conceptually if they are authored by authors from non-affiliated organizations. In a pre-processing step 212, a subset of the pages crawled by a search engine are identified as experts (for example, 2.5 million of 140 million or so pages might be found to be experts). The pages in this subset are indexed in a special inverted index called an expert reverse index. A. Creating an Expert Reverse Index After receipt of an input query 214, a lookup 216 is done on the expert reverse index to find and rank matching "expert pages." This phase computes the best expert pages on the query topic and as well as associated match information.	

				Then a target ranking 218 is performed. In this embodiment, match information is defined as the location(s) where query terms occur within the expert page." '604 Patent at 4:65-5:40: "a) Determining Page Affiliation Two pages are "affiliated" if and only if the hosts they are located on are known to be affiliated. Thus, as shown in FIG. 10, a determination about whether two pages are affiliated is made by looking at the hosts of the pages. Two hosts are affiliated if one or both of the following is true: (i) They share the same first 3 octets of the IP address, and (ii) The rightmost non-generic token in the hostname is the same. In the described embodiment, only suffixes beginning with a period "." are considered. A suffix is considered generic if it occurs in a large number of distinct hosts. E.g., ".com" and ".co.uk" are domain names that occur in a large number of hosts and are hence generic suffixes. Given two hosts, if the generic suffix in each case is stripped and the subsequent right-most token is the	

				same, they are considered to be affiliated. Tokens are substrings of the hostname delimited by ".". For example, in comparing "www.ibm.com" and "ibm.co.mx," the generic suffixes ".com" and ".co.mx" are ignored. The resulting rightmost token is "ibm," which is the same in both cases. Hence they are considered to be affiliated. Optionally, the generic suffix could be required to be the same in both cases. The affiliation relation is transitive: if A and B are affiliated and B and C are affiliated then we take A and C to be affiliated even if there is no direct evidence of the fact. In practice some non-affiliated hosts may be classified as affiliated, but that is usually acceptable since this relation is intended to be conservative. Preprocessing step 202 of FIG. 2(a) preferably constructs a host-affiliation lookup. Using a standard union-find method, hosts are grouped that either share the same rightmost non-generic suffix or have an IP address in common, into sets. Every set is given a unique identifier (e.g., the host with the lexicographically lowest hostname). The host-affiliation lookup maps every host to its set identifier or to itself (when there is no set). This is used to	

				compare hosts. If the lookup maps two hosts to the same value then they are affiliated; otherwise they are non-affiliated. Thus, in the described embodiment, the test for affiliation can be done very quickly." FIGS. 2(a) and 10. '604 Patent at 5:41-61: "b) Finding Expert Pages Step 202 preferably processes a search engine's database of pages and selects a subset which is considered to be good sources of links on specific topics, albeit unknown. As shown in element 302 of FIG. 3(a), all pages with out-degree greater than a threshold, k (e.g., k=5) are considered. The out-degree is the number of out-going links from the page. In element 306, if the URLs of a such a page point to k distinct non-affiliated hosts that are mutually non-affiliated as well, the page is an expert page (see element 308). In the described embodiment, all selected expert pages are downloaded from the web if they are not already available. FIG. 3(b) shows an alternate method of determining expert pages. As shown in element 317 of FIG. 3(b), if a broad classification (such as Arts, Science,	

				Internet etc.) is known for every page in the search engine database then we can additionally require that all the k non-affiliated URLs discovered in the previous step point to pages that share the same broad classification." FIGS. 3(a) and 3(b). '604 Patent at 6:37-7:17: "B. Ranking Experts For an expert to be useful in response to a query, the minimum requirement is that there is at least one URL which contains all the query keywords in the key phrases that qualify it. A fast approximation would be to require all query keywords to occur in the page. Thereupon the an Expert Score of the expert is computed as a 3-tuple of the form (S_0, S_1, S_2). FIG. 8 is a flow chart of a method of ranking expert pages in accordance with a current query. Element 802 shows how to determine a level score for a current phrase on a current expert page. LevelScore(p) is a score assigned to the phrase by virtue of the type of phrase it is. E.g., we could use a LevelScore of 16 for title phrases,	

				6 for headings and 1 for anchor text. This is based on the assumption that the title text is more useful than the heading text, which is more useful than an anchor text match in determining what the expert page is about. Element 804 shows how to determine a fullness factor for a current phrase (p) on a current expert page for the query FullnessFactor(p,q) is a measure of the number of terms in p covered by the terms in q. Let plen be the length of p. Let m be the number of terms in p which are not in q (i.e., surplus terms in the phrase). If $m<=2$,FullnessFactor(p,q)=1 If $m>2$,FullnessFactor(p,q)=$1+(2-m)/plen$ Element **806** shows how to determine the expert score for a current expert page. Let k be the number of terms in the input query, q. The component S_i of the score is computed by considering only key phrases that contain precisely $k-i$ of the query terms. E.g., S_0 is the score computed from phrases containing all the query terms.	

				$S_i=\Sigma_{(\text{key phrases p with k}-\text{i query terms})}\text{LevelScore}(p)*\text{FullnessFactor}(p,q)$ The score of the expert can be converted into a scalar by determining a weighted sum of the components. E.g., $\text{ExpertScore}=2^{32}*S_0+2^{16}*S_1+S_2$. In other words, S_0, S_1, and S_2 can be stored, for example, in respective bytes in memory to form a scalar. Elements **808**, **810**, **812**, and **814** indicate that each keyword an expert page is considered when determining the expert score for that page. An expert score is determined for each expert page. Element **816** ranks experts in accordance with their expert scores, which, in the described embodiment, are formed from scores S_0, S_1, and S_2." Krishna Bharat and George A. Mihaila, *Hilltop: A Search Engine based on Expert Documents*: "We define an expert page as a page that is about a certain topic and has links to many non-affiliated pages on that topic. . . . We felt than an expert page needs to be objective and diverse: that is, its recommendations should be unbiased and point to numerous *non-affiliated* pages on the subject. Therefore, in order	

				to find the experts, we needed to detect when two sites belong to the same or related organizations. . . . The key difference consists in the fact that we are only considering "expert" sources - pages that have been created with the specific purpose of directing people towards resources. In response to a query, we first compute a list of the most relevant experts on the query topic. Then, we identify relevant links within the selected set of experts, and follow them to identify target web pages. The targets are then ranked according to the number and relevance of non-affiliated experts that point to them. Thus, the score of a target page reflects the collective opinion of the best independent experts on the query topic" '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000885-00000890: "Appellants' invention relates to a computer search engine for searching a large number of hypertext documents (Specification 1:5-6). In response to an input query, the search engine ranks matching expert pages (pages about a topic with links to several non-affiliated pages) . . . ." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-	

				00000910-00000937: "Chakrabarti never teaches forming a set of expert documents from all hypertext documents without reference to a search query. Chakrabarti does not teach ranking expert documents, but instead ranking a topic based subset of documents." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000942-00000963: "In fact, Chakrabarti does not teach forming a set of experts at all. Chakrabarti only teaches ranking of subsets of documents that are produced in topic based searches, i.e., ranking documents that relate to a particular topic." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00001013-00001030: "The first phase may also include indexing of the expert list by topics to create an expert reverse index. . . . This step also occurs after the experts have been identified, but before a topic based query is received." VALTRUS-GOOGLE-NDTX-00000967-00000971. VALTRUS-GOOGLE-NDTX-00000992-00000996.	

				U.S. Patent 6,862,710, cited on face of '604 Patent, at 4:53-5:9: "While many of the above techniques improve search results based on previous user's preferences, none attempts to interpret word meaning or overcome the fundamental problems of synonymy, polysemy and search by concept. These are addressed by expert systems consisting of electronic thesauri and lexical knowledge bases. The design of a lexical knowledge base in existing systems requires the involvement of a large teams of experts. It entails manual concept classification, choice of categories, and careful organization of categories into hierarchies (Bateman et al, 1990, No. 3 in Appendix A; Bouad et al, 1995, No. 7 in Appendix A; Guarino, 1997, No. 14 in Appendix A; Lenat and Guha, 1990, No. 20 in Appendix A; Mahesh, 1996, No. 23 in Appendix A; Miller, 1990, No. 25 in Appendix A; Mahesh et al, 1999, No. 24 in Appendix A; Vogel, 1997 and 1998, Nos. 31 and 32 in Appendix A). In addition, lexical knowledge bases require careful tuning and customization to different domains. Because they try to fit a preconceived logical structure to a collection of documents, lexical knowledge bases typically fail to deal effectively with heterogeneous collections such as the Web. By	

				contrast, the approach known as Latent Semantic Indexing (LSI) uses a data driven solution to the problem of lexical categorization in order to deduce and extract common themes from the data at hand." U.S. Patent 6,728,752, cited on face of '604 Patent, at 3:26-36: "Knowledge Pump, a Xerox system, provides community-based recommendations by initially allowing users to identify their interests and "experts" in the areas of those interests. Knowledge Pump is then able to "push" relevant information to the users based on those preferences; this is accomplished by monitoring network traffic to create profiles of users, including their interests and "communities of practice," thereby refining the community specifications. However, Knowledge Pump does not presently perform any enhanced search and retrieval actions like the search-engine-based systems described above." U.S. Patent 6,728,752, cited on face of '604 Patent, at 35:14-22: "The most popular pages in the nearest cluster can then be identified (step 2524) and recommended to the new user (step 2526). In an alternative embodiment of the invention, the names, e-mail addresses, or other	

				identifying data for the users in the nearest cluster (or at least one user in that nearest cluster, identified via the aggregate cosine similarity metric described above) can be provided to the new user, thereby allowing the new user to identify "experts" in a desired area."	
'604	1, 13, 14, 15, 16	target document(s) [cl. 1: pointed to by the ranked expert documents]	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification; the claims explain that target documents are documents "pointed to by" expert documents.	'604 Patent at 7:44-50: "In element 902 of FIG. 9 we consider the top N experts in the ranking from the previous step (for example, N=200) and consider the pages they point to. These pages are called targets." FIG. 9. '604 Patent at 3:3-9: "Next, target ranking looks at the out-going links from identified expert pages. By combining relevant out-going links from many experts on the query topic, it is possible to find the pages that are most highly regarded by the community of pages related to the query topic. This is the basis of the high relevance that the described embodiment of the invention delivers." '604 Patent at 3:16-18: ". . . ranking target documents pointed to by the ranked expert documents; and returning a results list based on the ranked target documents."	

				'604 Patent at 3:26-29: ". . . a software portion configured to rank target documents pointed to by the ranked expert documents; and a software portion configured to return a results list based on the ranked target documents." '604 Patent at 7:30-43: "Given the top ranked matching expert pages and associated match information determined by the method of FIG. 8, we select a subset of the hyperlinks within the expert pages. (Associated match information is preferably information about the key phrases within the expert pages that match query terms.) Specifically, we select links that we know to have all the query terms associated with them. This implies that the link matches the query. With further connectivity analysis on the selected links we identity a subset of their targets as the top-ranked pages on the query topic. The targets we identify are those that are linked to by at least two non-affiliated expert pages on the topic. The targets are ranked by a ranking score which is computed by combining the scores of the experts pointing to the target." '604 Patent at 8:13-29: "The approach described above generates a list of target pages which are likely to be very	

				authoritative pages on the topic of the query. This is by virtue of the fact that they are highly valued by pages on the WWW which address the topic of the query. In computing the value of a target page from the hyperlinks pointing to it, we only consider links originating from pages that seem to be experts. Specifically we require them to point to many non-affiliated sites. This is an indication that these pages were created for the purpose of directing users to resources, and hence we regard them as experts. Additionally, to boost relevance, we require a match between the query and the text on the expert page which qualifies the hyperlink being considered. This insures that hyperlinks being considered are on the query topic. The result of the steps described above is to generate a listing of pages that are highly relevant to the user's query and of high quality, which is the goal of our invention." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00001013-00001030: "The second phase then includes ranking target documents identified by the small set of expert documents identified above. . . . The target document set is dramatically smaller than the set of all documents on the web which may include the query	

				topic. But the target documents are qualified by having been identified, pointed to, by the expert documents identified in the first phase of the process." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000885-00000890: ". . . the search engine . . . looks at the targets (out-going links) from the expert pages) . . . ." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000910-00000937: "Then, target documents pointed to by the ranked experts are ranked and results based on the ranking of the target documents are returned." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000942-00000954: "The second phase then includes ranking target documents identified, i.e., pointed to, by the small subset of expert documents identified above as relating to the topic." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000942-00000963: "The second phase then includes ranking target	

				documents identified, i.e., pointed to, by the small subset of expert documents identified above as relating to the topic. . . . But the target documents are qualified by having been identified, pointed to, by the expert documents . . . ." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00001102-00001114: "The present invention then uses the ranked expert pages which are relevant to the topic to expand the list of target documents by ranking the pages to which the relevant experts point." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000967-00000971. '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000986-00000996. '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00001033-00001050. '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00001139-00001148.	

'604	1	without reference to the search query	A query is a sequence of words input by the user that produces a ranked list of URLs that are returned as search results.	'604 Patent at 1:23-30: "One solution to the problem of finding information of the web is to let software programs perform the search. Various search engines have been developed that return a list of ranked documents in response to a search query. If the query is broad (i.e., it matches many documents) then the returned list is usually too long for the user to look at fully. Users typically look only at the top ranked results on the assumption that they are most relevant." '604 Patent at 2:49-52: "The described embodiment of the present invention takes an input query, which is, for example, a sequence of words input by the user, and produces a ranked list of URLs that are returned as search results." '604 Patent at 2:65-3:2: "For a given input query, a lookup is done on the expert reverse index to find and rank matching expert pages. This phase computes the best expert pages on the query topic, as well as associated match information. The pages are ranked according to the match information." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000885-00000890: "Thus, the	

				formation of the set of expert documents occurs prior to a search query for a specific topic, but the ranking of the expert documents occurs after the search query. . . . 'Broad topic' [referred to by Chakrabarti] suggests something other than a specific search query." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000942-00000963: "That is, Chakrabarti starts his process after receiving a query with search terms. . . . Therefore, all of the searching and ranking taught by Chakrabarti includes or is based on a specific topic, i.e., a search query."	
'604	1	ranking target documents pointed to by the ranked expert documents	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'604	1, 19	ranking the expert documents in accordance with the search query	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		

EXHIBIT A – Valtrus's Proposed Constructions and Intrinsic and Extrinsic Evidence

'604	1, 2, 3, 4, 18	hypertext documents	Hypertext documents include pages and sites on the world wide web.	'604 Patent at 4:26-44: "FIG. 2(a) is a block diagram giving an overview of a preferred embodiment of the present invention. In the described embodiment, an initial set of hypertext pages 202 is obtained by, for example, a crawl of the world wide web. The hypertext documents (e.g., pages) are processed to yield a set of expert documents 204. . . . FIG. 2(b) is a flow chart of a search method in accordance with a preferred embodiment of the present invention. In the following example, the hypertext documents are pages (or sites) in the world wide web. It should be understood that the present invention can also be applied to other types of hypertext linked documents, such as hypertext databases." FIGS. 2(a) and 2(b). '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00001056-00001069: "Hypertext documents are pages in [the World] Wide Web."	"Hypertext." Alan Freedman, Computer Desktop Encyclopedia 440 (9th ed. 2001). VALTRUS-GOOGLE-NDTX-00007401-00007415. "Hypertext." Dictionary of Computer Science, Engineering, and Technology 232 (Phillip A. Laplante ed., 2nd ed. 2001). VALTRUS-GOOGLE-NDTX-00007416-00007434. "Hypertext." Steven M. Kaplan, Wiley Electrical and Electronics Engineering Dictionary 354 (2004). VALTRUS-GOOGLE-NDTX-00007435-00007455. "Hypertext." Rudolf F. Graf, Modern Dictionary of Electronics 357 (1999). VALTRUS-GOOGLE-NDTX-00007456-00007470. "Hypertext." S.M.H. Collin, Dictionary of Computing 163 (5th ed. 2004). VALTRUS-GOOGLE-NDTX-00007471-00007482. "Hypertext." Microsoft Computer Dictionary 261 (Alex Blanton and Sandra Haynes, eds., 5th ed. 2002). VALTRUS-GOOGLE-NDTX-00007483-00007498. "Hypertext." Harry Newton, Newton's Telecom Dictionary 473 (24th ed. 2008).

					VALTRUS-GOOGLE-NDTX-00007499-00007516. "Hypertext." Dick Pountain, The New Penguin Dictionary of Computing 234 (2001). VALTRUS-GOOGLE-NDTX-00007517-00007536. "Hypertext." Collins English Dictionary 802 (7th ed. 2005). VALTRUS-GOOGLE-NDTX-00007568-00007587. "Hypertext." The New Oxford American Dictionary 833 (Erin McKean ed., 2nd ed. 2005). VALTRUS-GOOGLE-NDTX-00007588-00007606. Tim Berners-Lee and Robert Cailliau, WorldWideWeb: Proposal for a HyperText Project (1990). VALTRUS-GOOGLE-NDTX-00007394-7400. John B. Smith et al., A Hypertext Writing Environment and its Cognitive Basis 195, *in* Hypertext '87 Papers (1987). VALTRUS-GOOGLE-NDTX-00007140-00007159. I. Ritchie, *HYPERTEXT – Moving Towards Large Volumes*, 32 THE COMPUTER J. 516, 516 (1989). VALTRUS-GOOGLE-NDTX-00007132-00007139.

'604	4	hypertext database	A database that comprises hypertext documents.		
'604	13, 17	edges	A connection from an expert page to a target page.	'604 Patent at 7:54-57: "1. As shown in element 906, for every expert E that points to target T we draw a directed edge (E,T). We compute an "edge score" for the edge (E,T) represented by EdgeScore(E,T) . . . ." '604 Patent at 7:63-67: "2. As shown in step 908, we next check for affiliations between expert pages that point to the same target. If two affiliated experts have edges to the same target T, we then discard one of the two edges. Specifically, we discard the edge which has the lower EdgeScore of the two." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000998-00001008: ". . . determining a plurality of edge scores for each target document, where an edge score is determined for edges between the expert document and the target document . . . ." '604 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00001121-00001135.	"Edge." Dictionary of Computer Science, Engineering, and Technology 154 (Phillip A. Laplante ed., 2nd ed. 2001). VALTRUS-GOOGLE-NDTX-00007416-00007434. Robin J. Wilson, Introduction to Graph Theory 8 (5th ed. 2010). VALTRUS-GOOGLE-NDTX-00007370-00007393. Po-Shen Loh, Graph theory 1 (2012). VALTRUS-GOOGLE-NDTX-00007286-00007292. Dawei Yin et al., Ranking Relevance in Yahoo Search (2016). VALTRUS-GOOGLE-NDTX-00007160-00007169. Maxie Inigo et al., College Mathematics for Everyday Life 197 (2021). VALTRUS-GOOGLE-NDTX-00007203-00007236.

'604	16	ES(E,T)	an edge score for the edge from an expert page E to a target page T	'604 Patent at 7:52-62: "The ranking score for a target T is computed in three steps: 1. As shown in element 906, for every expert E that points to target T we draw a directed edge (E,T). We compute an 'edge score' for the edge (E,T) represented by EdgeScore(E,T), which is computed thus: Let occ(k) be the number of distinct key phrases of expert E, within which a query keyword k occurs. If occ(k) is 0 for any query keyword then the EdgeScore(E,T)=0. Otherwise, EdgeScore(E,T)=ExpertScore(E)*$\Sigma_{(\text{query keywords } k)}$occ(k)"	
'604	17	edges to the same target	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'604	17	edge having a lower edge score	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		

'604	18	rightmost non-generic suffix	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'604	19	level score	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		

IV. U.S. Patent No. 6,816,809

Patent	Claim(s)	Term	Valtrus's Proposed Construction	Examples of Intrinsic Evidence	Examples of Extrinsic Evidence
'809	1, 13, 18	A *hardware based utilization metering device* / A *hardware based method for measuring processor utilization* in a computer system comprising a	A hardware-based device or method that measures when a processor is not executing any process.	'809 Patent at 1:49-61: "What is disclosed is a hardware based utilization metering device for use in a computer system having one or more central processor units (CPUs), the device comprising a state indicator coupled to a CPU, wherein the state indicator receives an indication when the CPU is in a first state; a counter coupled to the state indicator and coupled to a system clock, wherein the counter receives a measure of system time from the system	HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878. "Hardware." Alan Freedman, Computer Desktop Encyclopedia 416 (9th ed. 2001). VALTRUS-GOOGLE-NDTX-00007401-00007415. "Hardware." Dictionary of Computer Science, Engineering, and Technology 219 (Phillip A. Laplante ed., 2nd ed.

		plurality of processors / An apparatus that provides *hardware based utilization metering* of central processor units (CPUs) in a computer system		clock and receives data related to the indication when the CPU is in the first state, and generates a counter value indicative of time the CPU is in the first state; and a data usage provider coupled to the counter, wherein the data usage provider is capable of providing the counter value." '809 Patent at 1:62-2:4: "Also disclosed is a hardware based method for measuring processor utilization in a computer system comprising a plurality of processors, the method comprising determining when any of the plurality of processors is busy (i.e., not idle); providing a busy indication to a counter associated with the busy processor; receiving at the counter a measure of computer system time; incrementing a counter value in the counter based on the received busy indication and an amount of computer system time that the processor is determined to be busy; and maintaining the counter value." '809 Patent at 2:5-16: "Finally, what is disclosed is an apparatus that provides hardware based utilization metering of CPUs in a computer system, comprising a plurality of CPUs. Associated with the CPUs is means for measuring computer system time. In addition, for each of the plurality of CPUs, the apparatus	2001). VALTRUS-GOOGLE-NDTX-00007416-00007434. "Hardware." Steven M. Kaplan, Wiley Electrical and Electronics Engineering Dictionary 332-333 (2004). VALTRUS-GOOGLE-NDTX-00007435-00007455. "Hardware." Rudolf F. Graf, Modern Dictionary of Electronics 338 (1999). VALTRUS-GOOGLE-NDTX-00007456-00007470. "Hardware." S.M.H. Collin, Dictionary of Computing 156 (5th ed. 2004). VALTRUS-GOOGLE-NDTX-00007471-00007482. "Hardware." Microsoft Computer Dictionary 246 (Alex Blanton and Sandra Haynes, eds., 5th ed. 2002). VALTRUS-GOOGLE-NDTX-00007483-00007498. "Hardware." Harry Newton, Newton's Telecom Dictionary 453 (24th ed. 2008). VALTRUS-GOOGLE-NDTX-00007499-00007516. "Hardware." Dick Pountain, The New Penguin Dictionary of Computing 220 (2001). VALTRUS-GOOGLE-NDTX-00007517-00007536.

				includes means for determining when the processor is busy; means for providing a busy indication when the processor is busy; means for receiving the busy indication and a measure of computer system time; means for combining the busy indication and the measure of computer system time to generate a counter value indicative of processor utilization; and means for maintaining the counter value." '809 Patent at 3:50-67: "FIG. 1A is a basic block diagram of a computer system 100 that implements hardware based utilization metering." FIG. 1A. '809 Patent at 4:66-5:17: "FIG. 1C shows selected components of the hardware based utilization metering device in more detail." FIG. 1C. '809 Patent at 7:3-31: "FIG. 4 is a flow chart showing a hardware based CPU utilization operation 600 using the system 100″ of FIG. 3. The operation 600 starts in block 610. In block 620, the cell 200 is replaced and the hardware components on the cell 200 are powered up. In block 630, the usage data provider 500 receives an indication	"Hardware." Collins English Dictionary 745 (7th ed. 2005). VALTRUS-GOOGLE-NDTX-00007568-00007587. "Hardware." The New Oxford American Dictionary 770 (Erin McKean ed., 2nd ed. 2005). VALTRUS-GOOGLE-NDTX-00007588-00007606.

				of the power up of the cell 200 components, and the usage data provider 500 sends the stored counter values 501-504 to the counters 240-243, respectively. In block 640, the received counter values are used to reinitialize the counters 240-243. However, the CPUs 210-213 are idle (operating systems are not processing), and in block 650, the idle indicators 220-223 provide an idle indication to their respective counters 240-243, thereby preventing incrementing of the counter values. In block 660, the CPU 210 begins running an operating system, and the idle indicator 220 sends an indication to the counter 240 that the CPU 210 is not idle. In block 670, the counter 240 receives the non-idle indication, and the system time from the system clock 230, and begins incrementing the counter value for the CPU 210. In block 680, the operating system running on the CPU 210 stops processing, halts the CPU 210, and asserts a halt indication. In block 690, the counter 240 receives an idle indication and stops incrementing the counter value. The operation 600 may continue with incrementing counter values for other CPUs in the system 100″ and may include routines to update	

				the non-volatile counter values in the usage data provider 500." FIG. 4. '809 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000795-00000802: "In contrast to Bishop, claim 1 as amended recites an idle indicator coupled to a processor. The idle indicator is a hardware device coupled to a pin of the processor and the hardware device reads a signal asserted on the pin when the processor is idle, i.e., not executing <u>any</u> process (*See* claim 3). . . . In contrast to Bishop and Boudreau, individually and in combination, claim 13 recites a hardware based method · for measuring process for utilization in a computer system comprising a plurality of processors." U.S. Patent 4,503,495, cited on face of '809 Patent, at 1:38-66: "The purpose of a hardware analyzer is to monitor and analyze the various aspects of the operation of the data processing system hardware. For example, a hardware analyzer may monitor various timings within the CPU or transfer of information between units connected to a bus. For example, the hardware analyzer may monitor the time that it	

				takes for the memory to respond to a CPU request for reading a word from memory with this time being the average time reflecting those cases in which the memory is immediately available for reading and those cases which the reading is deferred because the memory is busy performing a data transfer to a peripheral device. Hardware analyzers are also used to determine utilization factors, for example, the percentage of time that the CPU is being utilized, as compared to the percentage of time the CPU is idle, waiting for either data from memory or the completion of an input/output operation. The hardware analyzer may also be used to determine utilization facts and response times for various components within the system, such as peripheral devices, and memory subsystems. The data provided by the hardware analyzer may be used in various ways. For example, the diagnosis of system design problems or the optimization of system configurations as a data processing system is either contracted or expanded by the addition or removal of equipment in response to optimizing system for an existing data processing workload or to accommodate a changing data processing workload."	

				European Patent EP0320329A2, cited on face of '809 Patent, at 3:26-40: "The present invention provides these and other advantageous features by including diagnostic instructions in the processor "idle loop." A processor does not cease performing instructions when it is not busy, but instead jumps or "traps" to a so-called "idle loop" whenever it is idle. The idle loop generally consists of instructions which perform no useful function (e.g., "no operation," delay and/or jump instructions). When the processor must perform a function, it receives an "interrupt" -- at which time it ceases performing instructions in the idle loop and begins performing other, useful program control instructions. The next time the processor has no further tasks to perform, it once again returns to its idle loop." U.S. Patent 5,654,905, cited on face of '809 Patent, at 1:44-50: "In accordance with another aspect of the present invention, the time logging program intercepts idle interrupts, which are issued by the computer's operating system whenever it is idle. The time logging program updates the log file only during these idle periods, thus further ensuring that the program's	

				operation does not interfere with the computer's other tasks." U.S. Patent 6,049,798, cited on face of '809 Patent, at 2:27-39: "CPU idle time in a data processing system is the amount of time the computer's Central Processing Unit (CPU) is not being utilized by any task. Previous methods for measuring CPU idle time used a thread to perform a series of tasks. The number of tasks the thread performed was then compared with a hypothetical number of tasks that could have been performed, if the thread was allowed all available CPU time. This procedure is lacking in that the hypothetical number of tasks is different on different data processing systems. A system specific calibration algorithm is required to determine the minimum time the task(s) required to execute. This calibration method can be unreliable and presents many practical problems when moving between systems."	
'809	1, 3, 4, 10	idle indicator	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.

'809	1, 5, 6, 7, 8, 10, 11, 13, 14, 15, 16, 17, 18, 19	counter	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.
'809	1	system clock	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.
'809	1, 7, 9	data usage provider	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.
'809	2, 5, 13, 18	busy / busy state / busy indication	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.
'809	12, 18	Cells	A set of components that can be removed from the computer system as a group.	'809 Patent at 5:42-54: "FIG. 3 is a block diagram of a computer system 100″ having multiple CPUs arranged in partitions or cells. One or more of the CPUs may run multiple instances of operating systems, or, certain CPUs	HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.

				may run specific instances of operating systems while other CPUs run other instances of operating systems. As shown in FIG. 3, the computer system 100″ includes hardware cells 1 though K, denoted as cells 200, 300, and 400, respectively. This arrangement is meant to indicate that the computer system 100″ may include K cells, where K is an integer, including 1 (one), indicating the computer system 100″ comprises one cell. Each of the cells is a set of components that can be removed from the computer system 100″ as a group." FIGS. 3 and 4. '809 Patent at 5:54-65: "As can be seen in FIG. 3, the arrangement of components within each cell is identical in terms of type and number of components. However, the cells need not include the same type or same number of components. In addition to division of components among the hardware cells 1 through K, the components may be partitioned, or logically sorted. The partitions may comprise any number of CPUs or any number of cells. For example, cell 1 (200) may comprise four separate partitions, one for each of the CPUs installed in cell 1. Alternatively, cell 1 (200) and cell 2 (300) may comprise a	

				first partition and the remaining cell(s) may comprise additional partition(s)." '809 Patent at 5:66-6:9: "Taking the cell 200 as an example, there are included four CPUs 210-213. However, the cell 200 may include more or fewer than four CPUs. Associated with each of the CPUs 210-213 is a corresponding idle indicator 220-223. Each of the idle indicators receives an indication when its associated CPU is idle, and provides an output to a corresponding counter 240-243 that provides a current counter value (CPU utilization metric). Each of the counters 240-243 receives an input from a system clock 230. The system clock 230 functions in the same manner as the system clock 130 shown in FIG. 1A." FIG. 1A. '809 Patent at 6:17-19: "The arrangement of the cells 200, 300, and 400 allows one or more of the cells to be removed from the computer system 100″ while maintaining the computer system in operation." '809 Patent at 6:30-39: "Should one of the cells 200, 300 or 400 be replaced with a new cell, or with the original cell, but with one or more new CPUs, the	

				counter values may be lost when the cell is powered off. Thus, whenever a cell is powered on, the usage data provider 500 will reinitialize the corresponding counters in the cell. For example, if the cell 200 is removed, and the CPU 210 is replaced with a new CPU 210′, then when the original cell 200 is reinstalled and powered on, the usage data provider 500 will provide the stored counter value 501 to the counter 240." '809 Patent at 7:4-12: "FIG. 4 is a flow chart showing a hardware based CPU utilization operation 600 using the system 100″ of FIG. 3. The operation 600 starts in block 610. In block 620, the cell 200 is replaced and the hardware components on the cell 200 are powered up. In block 630, the usage data provider 500 receives an indication of the power up of the cell 200 components, and the usage data provider 500 sends the stored counter values 501-504 to the counters 240-243, respectively." FIG. 4. '809 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00000795-00000802: "Furthermore, Boudreau discloses only one CPU, not a	

				plurality of processors arranged into one or more cells as asserted in the Office Action at page 4. In particular, the items 301, 322, 320 of Figure 1 noted by the Examiner are actually parts of the software analyzer and not one of a plurality of processors arranged in one or more cells. In fact, nothing in Boudreau discloses use of more than one processor or having more than one processor arranged in multiple cells."	

V. U.S. Patent No. 7,523,454

Patent	Claim(s)	Term	Valtrus's Proposed Construction	Intrinsic Evidence	Extrinsic Evidence
'454	1, 5, 17	partitioned server	a single server or aggregation of server resources subdivided to perform as multiple servers.	'454 Patent at 1:34-36: "A partitioned server is a single server or aggregation of server resources subdivided to perform as multiple servers." '454 Patent at 1:36-38: "Thus, a partitioned server continues to service multiple locations, multiple departments, and/or multiple transactions." '454 Patent at 2:4-14: "The server may include a logical partition. That is, the server may be a consolidation of	HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001871-00001874.

				multiple, independent servers, each having its own resources, operating system, etc., which make up the partitions thereon. Alternatively, the server may include resource partitions. That is, the server may be a single server having separate resources allocated to each partition. Furthermore, the partitions may be statically or dynamically configured. In addition, the partitioned server may be included in a pool of servers including other partitioned and/or non-partitioned servers." '454 Patent at 2:43-67: "The method of the invention may comprise identifying a plurality of partitions on the server, determining a configuration of each partition, optionally determining at least one characteristic of the transaction, and providing the configuration to a load balancer, wherein the load balancer routes the transaction to one of the partitions based at least in part on the configuration thereof, and optionally also based on the characteristic(s) of the transaction. . . . As such, the apparatus and method of the invention recognizes and routes the transaction to the partition on the server, based at least in part on the configuration of the partition. The configuration also preferably accounts for the relative	

				weight of each partition and whether the server is statically and/or dynamically partitioned." '454 Patent at 4:13-40: "It is further understood that the server 140 may be any suitable server and may comprise any number of partitions 160-162. Preferably, the server 140 comprises logical partitions 160-162. That is, the server 140 comprising logical partitions 160-162 represent independent resources in a single, multiprocessing system having independent resources (e.g., CPU, memory, etc.). For example, a server with logical partitions 160-162 may include three processors, each processor identified as a separate partition 160 and running a separate operating system thereon. Logical partitioning is available, as an example, for IBM AS/400e series systems with multiple processors. Alternately, the server 140 comprising resource partitions 160-162 may be a single server having separately allocated resources. For example, a server 140 with resource partitions 160-162 may have a single processor that is allocated 20% to a first partition, 30% to a second partition, and 50% to a third partition. Although there is only a single processor with a single operating system, each of the three partitions	

				appears on the network (e.g., via IP address) as three distinct servers. Resource partitioning is available, as an example, using the Process Resource Manager available from Hewlett Packard Company. Also as an example, see co-owned U.S. patent application Ser. No. 09/493,753 filed Jan. 28, 2000 and titled DYNAMIC MANAGEMENT OF COMPUTER WORKLOADS THROUGH SERVICE LEVEL OPTIMIZATION, hereby incorporated by reference for all that it discloses. However, it is understood that the server 140 may be partitioned in any suitable manner." '454 Patent at 4:41-64: "Furthermore, the partitions 160-162 may be statically configured. That is, statically configured partitions 160-162 comprise resources or resource allocations that do not change during the operation thereof. For example, statically configured partitions 160-162 may be allocated two CPUs, or may be allocated 50% of a single CPU at startup or initialization until the server 140 is shut down, taken offline, or otherwise reinitialized. Alternatively, the partitions 160-162 may be dynamically configured. That is, dynamically configured partitions 160-162 comprises resources and/or resource allocations that may change	

				during the operation thereof. For example, dynamically configured partitions 160-162 may also be allocated two CPUs, or may be allocated 50% of a single CPU at startup or initialization. However, the configuration of the partitions 160-162 may change during the operation of the server 140. For example, the partitions 160-162 may be allocated an additional CPU, or 75% of a single CPU. In yet another embodiment, one or more partitions 160-162 may be statically configured, while other partitions 160-162 are dynamically configured. In addition, the partitions 160-162 may be dedicated (e.g., to a corporate department, for a particular purpose, etc.). In such an embodiment, the dedication may be included as part of the configuration of the partitions 160-162." '454 Patent at 5:49-58: "Alternatively, the transaction 110 is shown in FIG. 2 routed to a partition F 160 of the server 140 that is selected from a server pool 200 on the network 120. In this embodiment, the transaction 110 may be routed 180 to a partition 160 on a single server 140, and/or routed 210 to another server 220 in the server pool 200. The other server 220 may be a partitioned server, in which the	

				transaction 110 is routed to a partition thereon, as described above. Alternatively, the other server 220 may be a non-partitioned server, in which the transaction 110 is routed to the server itself." FIG. 2. '454 Patent at 8:13-20: "FIG. 4 illustrates one embodiment of a method for routing the transaction to the server. In step 400, the partitions 160-162 are identified on the server 140. For example, the agent 170 obtains the partition identifications (e.g., 310 in FIG. 3) and the network location (e.g., network address 320 in FIG. 3) for each partition 160-162 on the server 140. The agent 170 may identify the partitions 160-162 through interaction with the partition manager 150, or otherwise, as discussed above." FIG. 4. '454 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00001569-00001572: "The load balancer of Applicants' claim 1 could therefore be used to balance a load between the partitions of a single partitioned server, between the partitions of different servers, or	

				between partitioned and non-partitioned servers." '454 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00001545-00001552: "Applicants believe it is inappropriate to read their 'partitions on a partitioned server' on Varma's partitions of a distributed server. Applicants define what they mean by a 'partitioned server' on page 2, lines 3-5, of their specification, which states, 'A partitioned server is a single server or aggregation of server resources subdivided to perform as multiple servers.' Thus, in the context of Applicants' specification and claims, a partitioned server does not comprise multiple servers, but rather a single server that 'perform[s] as multiple servers.'" '454 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00001426-00001441: "A significant distinction between the Aman reference and Applicants' invention is that Aman teaches a method of assigning work to one or more servers, whereas the Applicants' invention discloses a method for routing a transaction to a partitioned server."	

				'454 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00001462-00001482. '454 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00001521-00001541: "However, it is noted that all of Appellants' claims recite 'routing a transaction to a partitioned server' rather than routing a transaction within a partitioned server. It is therefore believed that the scope and context of Appellants' claims is clear, and that Appellants' arguments regarding the differences between their claims and Zalewski's brief mention of 'load balancing of the 1/0 workload' is relevant. Obviously, there must be some way to route a transaction received by a partitioned server to one of the partitions within the partitioned server. However, what the art was lacking was a way to route a transaction to a partition of a partitioned server. This is what is set forth, in different ways, in Appellants' claims." U.S. Patent 6,542,926, cited on face of '454 Patent, at Abstract: "Multiple instances of operating systems execute cooperatively in a single multiprocessor computer wherein all processors and resources are electrically connected together. The single physical machine	

				with multiple physical processors and resources is subdivided by software into multiple partitions, each running a distinct copy, or instance, of an operating system. Each of the partitions has access to its own physical resources plus resources designated as shared. The partitioning is performed by assigning all resources with a configuration tree. None, some, or all, resources may be designated as shared among multiple partitions. Each individual operating instance will generally be assigned the resources it needs to execute independently and these resources will be designated as "private." Other resources, particularly memory, can be assigned to more than one instance and shared. Shared memory is cache coherent so that instances may be tightly coupled, and may share resources that are normally allocated to a single instance. This allows previously distributed user or operating system applications which usually must pass messages via an external interconnect to operate cooperatively in the shared memory without the need for either an external interconnect or message passing. Examples of application that could take advantage of this capability include distributed lock managers and cluster interconnects. Newly-added resources, such as CPUs and memory,	

				can be dynamically assigned to different partitions and used by instances of operating systems running within the machine by modifying the configuration." U.S. Patent 6,542,926, cited on face of '454 Patent, at 2:55-67: "The VM architecture supports the concept of a "logical partition" or LPAR. Each LPAR contains some of the available physical CPUs and resources which are logically assigned to the partition. The same resources can be assigned to more than one partition. LPARs are set up by an administrator statically, but can respond to changes in load dynamically, and without rebooting, in several ways. For example, if two logical partitions, each containing ten CPUs, are shared on a physical system containing ten physical CPUs, and, if the logical ten CPU partitions have complementary peak loads, each partition can take over the entire physical ten CPU system as the workload shifts without a re-boot or operator intervention." U.S. Patent 6,542,926, cited on face of '454 Patent, at 3:48-62: "Hive cells are not responsible for deciding how to divide their resources between local and remote requests. Each cell is responsible only for maintaining its internal	

				resources and for optimizing performance within the resources it has been allocated. Global resource allocation is carried out by a user-level process called "wax." The Hive system attempts to prevent data corruption by using certain fault containment boundaries between the cells. In order to implement the tight sharing expected from a multiprocessor system, despite the fault containment boundaries between cells, resource sharing is implemented through the cooperation of the various cell kernels, but the policy is implemented outside the kernels in the wax process. Both memory and processors can be shared." U.S. Patent 6,542,926, cited on face of '454 Patent, at 4:42-53: "In accordance with the principles of the present invention, multiple instances of operating systems execute cooperatively in a single multiprocessor computer wherein all processors and resources are electrically connected together. The single physical machine with multiple physical processors and resources is subdivided by software into multiple partitions, each with the ability to run a distinct copy, or instance, of an operating system. Each of the partitions has access to its own physical resources plus resources designated as shared. In	

				accordance with one embodiment, the partitioning is performed by assigning resources using a configuration data structure such as a configuration tree." U.S. Patent 6,542,926, cited on face of '454 Patent, at 10:38-51: "The hardware components are represented by a hardware root node 304 which contains children that represent a hierarchical representation of all of the hardware currently present in the computer system. "Ownership" of a hardware component is represented by a handle in the associated hardware node which points to the appropriate software node (310, 312 or 314.) These handles are illustrated in FIG. 4 which will be discussed in more detail below. Components that are owned by a specific partition will have handles that point to the node representing the partition. Hardware which is shared by multiple partitions (for example, memory) will have handles that point to the community to which sharing is confined. Un-owned hardware will have a handle of zero (representing the tree root node 302)." U.S. Patent 6,542,926, cited on face of '454 Patent, at 17:16-31: "A memory controller node (such as nodes 336 or 350) is used to express a	

				physical hardware component, and its owner is typically the partition which will handle errors, and initialization. Memory controllers cannot be assigned to communities, as they require a specific operating system instance for initialization, testing and errors. However, a memory description, defined by a memory descriptor node, may be split into "fragments" to allow different partitions or communities to own specific memory ranges within the memory descriptor. Memory is unlike other hardware resources in that it may be shared concurrently, or broken into "private" areas. Each memory descriptor node contains a list of subset ranges that allow the memory to be divided among partitions, as well as shared between partitions (owned by a community)." U.S. Patent 6,542,926, cited on face of '454 Patent, at 18:36-49: "FIG. 4 illustrates the configuration tree shown in FIG. 3 when it is viewed from a perspective of ownership. The console program for a partition relinquishes ownership and control of the partition resources to the operating system instance running in that partition when the primary CPU for that partition starts execution. The concept of "ownership" determines how the hardware resources	

				and CPUs are assigned to software partitions and communities. The configuration tree has ownership pointers illustrated in FIG. 4 which determine the mapping of hardware devices to software such as partitions (exclusive access) and communities (shared access). An operating system instance uses the information in the configuration tree to determine to which hardware resources it has access and reconfiguration control." U.S. Patent 6,542,926, cited on face of '454 Patent, at 21:24-29: "4. An instance may need to be capable of supporting multiple arbitrary physical holes in its address space, if it is part of a system configuration in which memory is shared between partitions. In addition, an instance may need to deal with physical holes in its address space in order to support 'hot inswap' of memory." U.S. Patent 6,778,540, cited on face of '454 Patent, at 8:7-22: "As noted, the OSA adapter is shared across any number of host partitions, and any one of these partitions could function as a router to forward packets from the adapter's LAN to another LAN to which the partition is connected. The purpose of a "set routing" command is to define	

				at the adapter a specific IP instance (i.e., partition) as a routing node. During operation, it is likely that inbound data frames will be received by the gateway (i.e., OSA adapter) destined to IP addresses that are not known to that system. These inbound frames may be routable by a TCP/IP instance known to the gateway to be the correct destination TCP/IP. In the IBM S/390 environment, there may be multiple system images, each with multiple TCP/IP instances. This presents the adapter with multiple issues, including, to which TCP/IP instance should the frames be routed, should the adapter route the frames to all known TCP/IP instances, and/or should the adapter discard the frames." U.S. Patent 7,051,188, cited on face of '454 Patent, at Abstract: "Allocation of shareable resources of a computing environment are dynamically adjusted to balance the workload of that environment. Workload is managed across two or more partitions of a plurality of partitions of the computing environment. The managing includes dynamically adjusting allocation of a shareable resource of at least one partition of the two or more partitions in order to balance workload goals of the two or more partitions."	

				U.S. Patent 7,051,188, cited on face of '454 Patent, at 4:30-40: "Central processors 106 are physical processor resources that are allocated to the logical partitions. In particular, each logical partition 108 has one or more logical processors (not separately shown for clarity), each of which represents all or a share of a physical processor 106 allocated to the partition. The logical processors of a particular partition 108 may be either dedicated to the partition (so that the underlying processor resource 106 is reserved for that partition) or shared with another partition (so that the underlying processor resource is potentially available to another partition)." U.S. Patent 7,051,188, cited on face of '454 Patent, at 5:20-40: "Coupling facility 122 (a.k.a., a structured external storage (SES) processor) contains storage accessible by the central processor complexes and performs operations requested by programs in the CPCs. The coupling facility is used by various aspects of the present invention for the sharing of state information used in making shared resource redistribution decisions. (In one embodiment, each central processor complex is coupled to a plurality of coupling facilities.) Aspects of the operation of a coupling	

				facility are described in detail in such references as Elko et al., U.S. Pat. No. 5,317,739 entitled "Method And Apparatus For Coupling Data Processing Systems", issued May 31, 1994; Elko et al., U.S. Pat. No. 5,561,809, entitled "In A Multiprocessing System Having A Coupling Facility Communicating Messages Between The Processors And The Coupling Facility In Either A Synchronous Operation Or An Asynchronous Operation", issued Oct. 1, 1996; Elko et al., U.S. Pat. No. 5,706,432, entitled "Mechanism For Receiving Messages At A Coupling Facility", issued Jan. 6, 1998; and the patents and applications referred to therein, all of which are hereby incorporated herein by reference in their entirety." U.S. Patent 7,051,188, cited on face of '454 Patent, at 6:46-55: "As examples, the resources to be shared include CPU resources, I/O resources, and memory, as well as co-processors or any other shareable resources the machine might provide. A particular logical partition group may or may not have access to all of the resources of a particular machine. In fact, multiple logical partition groups could be defined to operate concurrently on a single machine. In order to manage	

				each logical partition group effectively, the resources that make up a particular logical partition group are effectively scoped to that group." U.S. Patent 6,336,134, cited on face of '454 Patent, at Abstract: "A method for building a locking, migration, dynamic clients, and dynamic partitions capable distributed server for a real-time collaboration session supports the synchronous creation and deletion of partitions by clients as well as the addition and withdrawal of clients during a current collaboration session. The method is based on history servers for providing a history of modifications so that a newly-added client can compute the current state of a shared workspace. The history servers cache and granularize intermediate modification sequences so that computation space and time are reduced. The method supports migrating partition server(s), history server(s), a creation/deletion server, and a collaboration server to different machines. Partition(s) can be dynamically locked and unlocked and, in an extension of this procedure, creation and deletion of partition(s) can be pre-announced and supported. Advanced dynamic-partitioning activities like splitting a partition,	

				merging partitions, shifting data from partition to partition are carried out naturally by locking the concerned partitions during the process of execution." U.S. Patent 6,336,134, cited on face of '454 Patent, at 3:15-4:16: "It is therefore an object of the present invention to provide a method for building a locking, migration, dynamic clients, and dynamic partitions capable distributed server for a real-time collaboration session. In order to take into account the changing nature of the communication network, wherein machines can be added, loaded, and removed dynamically, the processes of the distributed server can be migrated from machine to machine, dynamically. Indeed, the distributed server processes can avoid dedicated server hosting machines by residing and running solely on the machines hosting the client processes. In this case, the distributed server processes can migrate as and when new client machines are added or removed. The extended distributed server supports locking a partition by a client whereby the client with a lock can modify the locked partition until the client releases the lock. According to the invention, there is provided a method for building a distributed server	

				capable of handling locking, migration, dynamic clients, and dynamic partitions for a real-time collaboration session. The method includes: (i) a method by which a collaboration front end or client can synchronously create some new partitions; (ii) a method by which a client can synchronously delete some existing partitions; (iii) a method by which a client can synchronously get permission for creating or deleting partitions; (iv) an optional method based on optional, application-supplied creation/deletion server that is an alternative for the method for getting permissions; (v) a method based on history servers for providing a history of modifications so that a newly-added client can compute the current state of a shared workspace; (vi) caching or granularizing of intermediate modification sequences by the aforesaid history servers so that computation space and time are reduced; (vii) a method each for migrating partition server(s), history server(s), a creation/deletion server, and a collaboration server; (viii) a method for locking partition(s) and unlocking partition(s) whereby partition servers can handle lock and unlock operations; (ix) an extension of the preceding method of locking and unlocking partition(s) to accept pre-announced	

				creation and deletion of partition(s); and (x) a method for carrying out advanced dynamic-partitioning activities like splitting a partition, merging partitions, shifting data from partition to partition—these activities are carried out naturally by locking the concerned partitions during the process of execution. With respect to the distributed server in application Ser. No. 09/241,991, these methods preserve: (a) generality—different definitions of a modification continue to be supported along with additional support for a locking capability; (b) extensibility—simple, comprehensive and easy-to-implement inter-partition synchronisation is provided; (c) generality for partition hierarchies and groups, wherein in addition to the capabilities in the distributed server disclosed in application Ser. No. 09/241,991, simultaneous or incremental creation or deletion, of any group of new or existing partitions is also supported (as applicable). In comparison to the centralized server and the distributed server disclosed in application Ser. No. 09/241,991, these methods further improve interoperability across heterogeneous software/hardware platforms by (a) extending the functionality of distributed-server-based real-time	

				collaboration to dynamic partitions and dynamic clients, and (b) allowing the distributed server processes to migrate and dynamically adapt to changing network (including frontends), and collaboration conditions, which in turn obviates the need for special support for immobile processes from the (hardware/software platforms/servers comprising the) network in the face of commonplace faults, transients, load relief, and balanced/unbalanced development of network load." U.S. Patent 6,336,134, cited on face of '454 Patent, at 6:11-7:31: "The extended distributed server of the present invention supports both static partitionings and dynamic partitionings. The support for dynamic partitionings provided by the extended distributed server includes not just creation and deletion of partitions, but also their merging and splitting. So in the partition examples described above, the dynamic-partitioning activities of merging and splitting of paragraphs, tables, and objects can be done besides their dynamic, isolated creation and deletion using the extended distributed server. The partitioning of a workspace needs to be explicitly provided to the distributed server in order for it to provide a distributed serialization	

				service for the partitions. The partitioning can be user supplied, in which case participant(s) including their assistant(s) explicitly deal with partitions and partition changes for collaborative sessions. Otherwise, the partitioning can be application supplied, in which case the partitioning is generated and maintained automatically by the application software associated with the shared workspace. Examples of this are applications that translate their work spaces into NSTP (Notification Service Transport Protocol) partitions such as places. For these applications, and for all other cases where workspace partitioning is made available, a distributed server can be plugged in as the serializing backend. So for example, synchrony in each NSTP partition can be maintained by the plugged-in distributed server. Partitionings provided to the distributed server can be hierarchical and/or grouped in nature. So for example, partitions can contain sub-partitions within them as in the case of hierarchical tables and 3D objects containing other 3D objects. Partitions and sub-partitions are treated simply as partitions by the distributed server. The distributed server presented here can support dynamically-created and dynamically-deleted hierarchical/grouped partitions. A	

				dynamically-created hierarchical/grouped partition can involve the simultaneous creation of multiple (sub)partitions, or the hierarchical/grouped partition can be built incrementally. Both of these cases are catered to by the distributed server. The corresponding cases for the dynamic deletion of a hierarchical/grouped partition are also supported by the distributed server. Here also, as in the invention disclosed in application Ser. No. 09/241,991, in order to use the distributed server, each work space modification request (e.g., a partition creation request) has to identify which partition(s) it can possibly affect, and therefore, which (remaining) partitions can remain oblivious of and therefore out of synchrony with the modification request. If a multiplicity of partitions is identified for a modification request, then the modification is treated as a compound modification over the identified partitions. If only one partition is identified as affected by a modification request, then the modification may be treated as an ordinary modification over the identified partition, although a compound modification over one partition can also be used. For the identified partition(s), the given	

				modification will be synchronized with respect to each of them by the distributed server. This means that for any partition that is one of the identified partitions, the position of the given modification in the sequence of modifications on the identified partition as seen by any collaboration client (that participates in the collaboration throughout the modification sequences) is the same. Furthermore, for every client, it is guaranteed that when the given modification is seen (i.e., processed) by the client, all modifications preceding the given modification on each of the identified partitions would have been processed by the client. Since each partition is either already there with an identical value for each client at the start of collaboration, or it gets created dynamically, wherein the creation is itself defined as a modification on the partition, processing of just the given modification implies that the state of the identified partitions becomes the same for every client. In other words, when clients process a given modification over one or more partitions, their states for the partitions get synchronized to be the same. Since this is true for all modifications of the workspace, processing of the workspace by its clients proceeds in a synchronous	

				manner when the distributed server is used."	
'454	1, 2, 3, 17, 18	configuration(s)	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001871-00001874.
'454	1, 4, 17, 18	partitions	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001871-00001874. Sajal K. Das et al., Latency Hiding in Dynamic Partitioning and Load Balancing of Grid Computing Applications 1 (2001). VALTRUS-GOOGLE-NDTX-00007918-00007925. K. Dussa et al., *Dynamic Partitioning in a Transputer Environment*, 90 ACM 203, 203 (1990). VALTRUS-GOOGLE-NDTX-00007943-00007953. Kee-Hyun Park & Lawrence W. Dowdy, *Dynamic partitioning of multiprocessor systems*, 18 INT'L J. OF PARALLEL PROGRAMMING 91, 91 (1989). VALTRUS-GOOGLE-NDTX-00008514-00008543.

'454	1	load balancer	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001871-00001874.
'454	1	rank	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001871-00001874.

VI. U.S. Patent No. 7,748,005

Patent	Claim(s)	Term	Valtrus's Proposed Construction	Examples of Intrinsic Evidence	Examples of Extrinsic Evidence
'005	8, 9	computing domain(s)	A protected division of a computer.	'005 Patent at 3:11-13: "The computers, particularly the large computers, or the network may be divided into protection domains or partitions." '005 Patent at 13:37-40: "Although some representative embodiments have been described in terms of allocating physical resources between partitions, representative embodiments may allocate resources between any suitable computing domain. Another suitable	

				computing domain is a virtual machine." '005 Patent at 13:59-14:20: "FIG. 7 depicts system 700 that allocates virtual resources according to one representative embodiment. System 700 may comprise a plurality of physical resources such as CPUs 701, memory 702, network interface card (NIC) 703, disk storage 704, and/or the like. System 700 includes host operating system 701. Host operating system 701 enables low level access to physical resources 701-704. Additionally, host operating system 701 includes a software layer that virtualizes the physical resources 701-704 to enable allocation of those resources to higher level software processes. The virtualization software layer may be implemented within the kernel of host operating system 701 as an example. System 700 further includes virtual machines 705-1 through 705-N. Virtual machines 705-1 through 705-N appear to software processes executing within the virtual machines to be physical server platforms. Virtual machines 705-1 through 705-N provide partition and isolation functionality. A software fault within any particular virtual machine 705 may only affect the respective	

				virtual machine 705, while software processes associated with the other virtual machines 705 may continue operations in an ordinary manner. Within virtual machines 705, respective guest operating systems (OS) 706-1 through 706-N may be executed. Additionally, one or several applications (shown as 707-1 through 707-N) may be executed within each virtual machines 705. Performance monitors 708-1 through 708-N generate performance data related to applications 707. The performance data may be gathered directly from applications 707 and/or from operating systems 706." FIG. 7. '005 Patent Prosecution History at VALTRUS-GOOGLE-NDTX-00001767-00001778: "The Applicants respectfully submit that Eilert does not show or suggest the feature of a computing domain comprising a respective second manager process which maintains a list comprising at least one level of priority for at least one application of the domain . . . ." U.S. Patent 6,330,586, cited on face of '005 Patent, at 15:41-50: "The terminal domain 101 "contains" the user and items owned or controlled by the user,	

				such as the user's personal profile and applications, and terminal. The Terminal Agent 102 is responsible for representing the terminal resources when interacting with agents in the retailer domain, 103, service provider domain, 104, and network provider domain, 106. The terminal resources normally include service applications (for example, an electronic mail editor), communication capabilities, and user-interface platforms." U.S. Patent 6,330,586, cited on face of '005 Patent, at 16:11-23: "In addition to agents, the service retailer domain 103 contains a set of management processes to allow facilities such as the following: to monitor activity, make configuration changes, specify and enforce policies, maintain quality, performance access etc. as demand and resources vary, enforce payment and security related mechanisms, assess interactions and add links to third parties, and advertise services. These management processes support the retail services provided and managed by the Access Agent 107, which services may be realised by resources supplied by the retailer such as billing, or may be realised by	

				resources supplied by agents in the network or service provider domains." U.S. Patent 7,203,941, cited on face of '005 Patent, at 4:11-23: "Implementations include an execution engine of a virtual machine that processes intermediate code to execute an application. The execution engine can exist on a variety of operating systems, each of which may or may not have virtual memory. The execution of the application creates a domain of the application. The domain of the application, in a virtual machine environment, is an isolation abstraction for each application in its domain. The executing application can initiate a request for the allocation of a native resource to the domain of the application. When the executing application requests a native resource, which usually occurs through operation of a library member of a class library, an allocation and collection routine is performed."	
'005	8	application priority levels	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		

'005	8	manager process	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'005	8	performance data	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'005	8, 11, 13	dynamically reallocating	Plain and ordinary meaning to a person of ordinary skill in the art in light of the specification.		
'005	9, 12	virtual machine(s)	A set of virtual resources created as software constructs.	'005 Patent at 13:40-43: "Another suitable computing domain is a virtual machine. For example, virtualization refers to the creation of virtual machines that coexist on one or several physical servers. Virtualization software typically executes in connection with a host operating system of the physical server. The virtualization software creates virtual resources as software constructs."	"Virtual machine." Alan Freedman, Computer Desktop Encyclopedia 1033 (9th ed. 2001). VALTRUS-GOOGLE-NDTX-00007401-00007415. "Virtual machine." Dictionary of Computer Science, Engineering, and Technology 522 (Phillip A. Laplante ed., 2nd ed. 2001). VALTRUS-GOOGLE-NDTX-00007416-00007434. "Virtual machine." Steven M. Kaplan, Wiley Electrical and Electronics

				'005 Patent at 14:5-9: "Virtual machines 705-1 through 705-N appear to software processes executing within the virtual machines to be physical server platforms. Virtual machines 705-1 through 705-N provide partition and isolation functionality." FIG. 7. U.S. Patent Application 2002/0169987 A1, cited on face of '005 Patent, at [0003]: "The typical computer system includes hardware (e.g., processor, keyboard, hard disk, floppy-disk, etc.) and operating-system software that runs on the processor to control the components of the computer system. A virtual machine monitor (VMM) is another software program that runs on the processor of the computer system to create a user-definable number of computing platform environments." U.S. Patent Application 2002/0169987 A1, cited on face of '005 Patent, at [0009]: "U.S. Pat. No. 5,850,449, entitled "SECURE NETWORK PROTOCOL SYSTEM AND METHOD," discloses a device for and a method of securely transmitting objects containing executable programs in place of conventional data packets. U.S. Pat. No. 5,850,449 implements its device	Engineering Dictionary 841 (2004). VALTRUS-GOOGLE-NDTX-00007435-00007455. "Virtual machine." S.M.H. Collin, Dictionary of Computing 351 (5th ed. 2004). VALTRUS-GOOGLE-NDTX-00007471-00007482. "Virtual machine." Microsoft Computer Dictionary 549 (Alex Blanton and Sandra Haynes, eds., 5th ed. 2002). VALTRUS-GOOGLE-NDTX-00007483-00007498. "Virtual machine." Harry Newton, Newton's Telecom Dictionary 992 (24th ed. 2008). VALTRUS-GOOGLE-NDTX-00007499-00007516. "Virtual machine." Dick Pountain, The New Penguin Dictionary of Computing 532 (2001). VALTRUS-GOOGLE-NDTX-00007517-00007536. Gerald J. Popek and Robert P. Goldberg, *Formal Requirements for Virtualizable Third Generation Architectures*, 17 COMMS. OF THE ACM 412, 413 (1974). VALTRUS-GOOGLE-NDTX-00007320-00007329. Keith Adams and Ole Agesen, A Comparison of Software and Hardware Techniques for x86 Virtualization

				and method by encrypting all transmissions, which the present invention does not. U.S. Pat. No. 5,850,449 uses a virtual machine module to execute platform independent programs (e.g., JAVA programs). The device and method of the present invention is not disclosed in U.S. Pat. No. 5,850,449. U.S. Pat. No. 5,850,449 is hereby incorporated by reference into the specification of the present invention." U.S. Patent 7,203,941, cited on face of '005 Patent, at 1:13-28: "When executing managed code in a virtual machine (VM) environment, an application can run on different platforms. In such VM environments, native resources are typically allocated to applications using system calls. These system calls can be made when executing code in the VM environment by callers from class libraries. These class libraries commonly call through to the native operating system to perform low level functionality, such as drawing and windowing management. When these system calls are made, native resources are allocated as an effect of the library call from the caller. These native resources must be kept in synchronization with their counterpart in a managed code portion of the VM.	(2006). VALTRUS-GOOGLE-NDTX-00007170-00007181. Michael Paleczny, Christopher Vick, and Cliff Click, The Java HotSpot Server Compiler, *in* Proceedings of the Java Virtual Machine Research and Technology Symposium (JVM '01) (2001). VALTRUS-GOOGLE-NDTX-00007307-00007319. Jeff Dike, A user-mode port of the Linux kernel 1, *in* Proceedings of the 4th Annual Linux Showcase & Conference, Atlanta (2000). VALTRUS-GOOGLE-NDTX-00007932-00007942. I. Peter Deutsch and Allan M. Schiffman, *Efficient Implementation of the Smalltalk-80 System*, 84 ACM 297, 297 (1983). VALTRUS-GOOGLE-NDTX-00007926-00007931. Tim Lindholm and Frank Yellin, The Java Virtual Machine Specification 2 (1997). VALTRUS-GOOGLE-NDTX-00007954-00008444.

				When the managed code portion of the VM is through using the native resources, it is desirable that these native resources be freed so that they can be allocated for use by other applications." U.S. Patent 7,433,951, cited on face of '005 Patent, at 2:51-65: "For example, in the context of virtual machine technology, several virtual machines (VM's) often run via respective virtual machine monitors (VMM's) on a common underlying hardware platform and must share physical resources. Each VM thus constitutes a guest system, and typically includes its own guest operating system. One or more applications then usually run in each guest. Among other functions, the VMM intercepts and converts requests for physical resources by the VM's into corresponding requests for actual hardware resources. In some systems, the VMM itself performs the functions of an operating system and is responsible for managing all the physical system resources such as CPU time, physical memory, and I/O bandwidth. In other systems, the VMM forwards the requests to the underlying host operating system."	

WHETHER PREAMBLES ARE LIMITING

I. '704 Patent Preambles

Claim(s)	Preamble	Position on Whether Preamble is Limiting
1	A method of merging result lists from multiple search engines, said method comprising:	Yes
12	A method of merging result lists from multiple search engines, said method comprising:	Yes

II. '764 Patent Preambles

Claim(s)	Preamble	Position on Whether Preamble is Limiting
1	A method of ranking search results, comprising:	Yes
7	A computer readable memory to rank search results, comprising:	Yes
14	A computer readable memory, comprising:	Yes

ALLEGED MEANS PLUS FUNCTION TERMS

The parties have identified certain terms as being governed by 35 U.S.C. § 112(f). Valtrus and Google agree that 35 U.S.C. § 112(f) applies to the terms below from U.S. Patent No. 6,816,809, and Valtrus identifies each structure, act, or material corresponding to each claim element below.

Google has further identified certain terms from U.S. Patent No. 6,738,764 as being governed by 35 U.S.C. § 112(f). None of these terms include the word "means," creating a presumption that 35 U.S.C. § 112(f) does not apply. *See Williamson v. Citrix Online, LLC*, 792 F.3d 1339, 1348-49 (Fed. Cir. 2015). Google has offered no evidence to overcome this presumption, and Valtrus does not consider these terms to be governed by 35 U.S.C. § 112(f). In the event the Court rules that the identified terms are governed by 35 U.S.C. § 112(f), Valtrus identifies additional structure, act, or material corresponding to each claim element below. But absent such a determination by the Court, Valtrus's position is that no further construction is needed at this time.

I. U.S. Patent No. 6,816,809

Claim(s)	Term	Structure	Function
18	means for measuring computer system time	A clock, and all equivalents thereof. '809 Patent at 1:53-55: "wherein the counter receives a measure of system time from the system clock." HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.	measuring computer system time

18	means for determining when a CPU is busy	Hardware or software, and all equivalents thereof. '809 Patent at 3:31-34: "The means for determining if the CPU is busy may include hardware means. Alternatively, the means for determining if the CPU is busy may include software means." '809 Patent at 3:40-44: "The hardware means for determining if the CPU is busy may include hardware modifications to the computer system. The alternate software means for determining if the CPU is idle may include modifications to the CPU's operating system(s)." HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.	determining when a CPU is busy
18	means for providing an indication when the CPU is busy	An idle indicator, and all equivalents thereof. '809 Patent at 3:57-58: "The idle indicator 120 is capable of providing either an "idle" indication or a "not-idle"/"busy" indication." '809 Patent at 4:36-56: "As noted above, the idle indicator 120 provides an indication that the CPU	providing an indication when the CPU is busy

		110 is not idle. The idle indicator may be implemented as a hardware modification to the computer system 100. [Two embodiments are described] In addition to the just-described two embodiments of the idle indicator 120, other structures and methods may be used to indicate the CPU 110 is idle." FIG. 1A. HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.	
18	means for receiving the indication and measure of computer system time	A counter, and all equivalents thereof. '809 Patent at 1:67-2:4: "receiving at the counter a measure of computer system time; incrementing a counter value in the counter based on the received busy indication and an amount of computer system time that the processor is determined to be busy" HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.	receiving the indication and measure of computer system time
18	means for combining the indication and the	A counter, and all equivalents thereof.	combining the indication and the measure of computer system time to generate a counter value indicative of CPU utilization

	measure of computer system time to generate a counter value indicative of CPU utilization	'809 Patent at 1:67-2:4: "receiving at the counter a measure of computer system time; incrementing a counter value in the counter based on the received busy indication and an amount of computer system time that the processor is determined to be busy" HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.	
18	means for maintaining the counter value	A usage data provider, and all equivalents thereof. '809 Patent at 4:12-14: "The usage data provider 150 tracks the counter value and maintains a non-volatile master copy of the counter value." '809 Patent at 4:19-23: "Because the usage data provider 150 maintains a non-volatile copy of the counter value, even if the CPU 110, or other hardware component is removed, in addition to a loss of power situation, an up-to-date, or nearly up-to-date value of the counter value is always available." HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.	maintaining the counter value

19	means for reinitializing the counter value upon power of the CPU	A usage data provider, and all equivalents thereof. '809 Patent at 4:14-23: "When the CPU 110 is powered on (or a hardware component containing the CPU 110 is powered on), the saved non-volatile counter value is provided from the usage data provider 150 to the counter 140 to initialize the counter value in the counter 140. Because the usage data provider 150 maintains a non-volatile copy of the counter value, even if the CPU 110, or other hardware component is removed, in addition to a loss of power situation, an up-to-date, or nearly up-to-date value of the counter value is always available." '809 Patent at 6:33-35: "Thus, whenever a cell is powered on, the usage data provider 500 will reinitialize the corresponding counters in the cell." HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.	reinitializing the counter value upon power of the CPU
20	means for reporting a CPU utilization value to a network	A usage data provider, and all equivalents thereof.	reporting a CPU utilization value to a network external to the computer system

	external to the computer system	'809 Patent at 4:24-33: "The usage data provider 150 maintains a connection, or network interface 160 to a system or network (not shown) that is external to the computer system 100. For example, the interface 160 may be a local area network (LAN) interface to a LAN. The LAN may include a management server that receives and processes information from the various computer systems coupled to the LAN, including the counter values that indicate CPU utilization. The usage data provider 150 can provide the current value of the counter value to the network by way of the network interface 160." '809 Patent at 4:31-33: "The usage data provider 150 can provide the current value of the counter value to the network by way of the network interface 160. The counter value may be provided periodically or when polled by the network." HP Invention Disclosure Form. VALTRUS-GOOGLE-NDTX-00001875-00001878.	

II. U.S. Patent No. 6,738,764

Claim(s)	Term	Structure (Means)	Function
7	a vector constructor to form a feature vector for each viewed document	Executable code stored in a memory, and all equivalents thereof. '764 Patent at 2:64-66: "Also attached to the system bus 26 is a memory 30, which may be primary and/or secondary memory. The memory stores a set of executable programs and related data." FIG. 1. '764 Patent at 3:37-39: "A vector constructor 70 operates on the document-query database 62 to produce a set of document vectors that are stored in a vector database 72."	form a feature vector for each viewed document
7, 14	a similarity processor to calculate a similarity score for said query utilizing the feature vector of said selected viewed document / a similarity processor to calculate a similarity score for	Executable code stored in a memory, and all equivalents thereof. '764 Patent at 2:64-66: "Also attached to the system bus 26 is a memory 30, which may be primary and/or secondary memory. The memory stores a set of executable programs and related data." FIG. 1.	calculate a similarity score for said query utilizing the feature vector of said selected viewed document / a similarity processor to calculate a similarity score for said query utilizing a feature vector that characterizes attributes and query words of a different query associated with said document

	said query utilizing a feature vector that characterizes attributes and query words of a different query associated with said document	'764 Patent at 3:48-53: "The memory 30 also stores a similarity processor 80. As discussed below, the similarity processor 80 calculates a similarity score between a query and a feature vector of a document. Thus, the similarity processor 80 populates a similarity database 82 with a set of similarity score entries 84."	